frontiers

Machine Learning Methods Applied to the Global Modeling of Event-Driven Pitch Angle Diffusion Coefficients During High Speed Streams

G. Kluth [1,2,*], **J.-F. Ripoll** [1,2], **S. Has** [3], **A. Fischer** [3], **M. Mougeot** [4] **and E. Camporeale** [5,6]

[1] *CEA, DAM, DIF, F-91297, Arpajon, France*
[2] *UPS, CEA, LMCE, F-91680, Bruyères-le-Châtel, France*
[3] *LPSM UMR 8001, Université de Paris, F-75013 Paris, France*
[4] *Centre Borelli UMR 9010, ENS Paris-Saclay, ENSIIE, F-91190 Gif, France*
[5] *CIRES, University of Colorado, Boulder, USA*
[6] *NOAA Space Weather Prediction Center, USA*

Correspondence*:
G. Kluth
gilles.kluth@cea.fr

ABSTRACT

Whistler-mode waves in the inner magnetosphere cause electron precipitation in the atmosphere through the physical process of pitch-angle diffusion. The computation of pitch-angle diffusion relies on quasi-linear theory and becomes time-consuming as soon as it is performed at high temporal resolution from satellite measurements of ambient wave and plasma properties. Such an effort is nevertheless required to capture accurately the variability and complexity of atmospheric electron precipitation, which are involved in various Earth's ionosphere-magnetosphere coupled problems. In this work, we build a global machine-learning model of event-driven pitch-angle diffusion coefficients for storm conditions based on the data of a variety of storms observed by the NASA Van Allen Probes. We first proceed step-by-step by testing 8 nonparametric machine learning methods. With them, we derive machine learning based models of event-driven diffusion coefficients for the storm of March 2013 associated with high-speed streams. We define 3 diagnostics that allow to highlight the properties of the selected model and to select the best method. Three methods are retained for their accuracy/efficiency: spline interpolation, the radial basis method, and neural networks (DNN), the latter being selected for the second step of the study. We then use event-driven diffusion coefficients computed from 32 high-speed stream storms in order to build for the first time a statistical event-driven diffusion coefficients that is embedded within the retained DNN model. We achieve a global mean event-driven model in which we introduce a two-parameter dependence, with both the Kp-index and time kept as in an epoch analysis following the storm evolution. The DNN model does not entail any issue to reproduce quite perfectly its target, i.e. averaged diffusion coefficients, with rare exception in the Landau resonance region. The DNN mean model is then used to analyze how mean diffusion coefficients behave compared with individual ones. We find a poor performance of any mean

models compared with individual events, with mean diffusion coefficients computing the general trend at best, due to their large variability.

The DNN-based model allows simple and fast data exploration of pitch-angle diffusion among its multiple variables. We finally discuss how to conduct uncertainty quantification of Fokker-Planck simulations of storm conditions for space weather nowcasting and forecasting.

——

Keywords: Pitch-angle diffusion, event-driven, machine learning, data exploration

1 INTRODUCTION

Pitch angle diffusion is one of the major mechanisms that drive the structure of the Van Allen radiation belts and cause the well-known two belt structure. Whistler-mode hiss waves are responsible for the scattering of energetic electrons by wave-particle interactions and their subsequent precipitation into the atmosphere, forming a region devoid of electrons in the inner magnetosphere, known as the slot region, between the two radiation belts (Lyons and Thorne, 1973). Observations of the dynamics of the slot from the NASA Van Allen Probes (Mauk et al., 2013) are, for instance, presented in Reeves et al. (2016). Radiation dose received by the electronics of orbiting spacecraft is then reduced in the slot region. In the atmosphere, Breneman et al. (2015) have observed a direct correlation between the pulsation of the whistler-mode hiss waves and precipitated electrons at $\sim$100 km observed from a balloon of the BARREL mission (Millan et al., 2013). Linking directly precipitations and wave activity remains an open research subject of the ionosphere-magnetosphere system (Millan et al., 2021). The recent review in Ripoll et al. (2020a) and references therein brings more insight on radiation belt physics and current open questions.

Pitch angle scattering can be computed either from statistical models derived from years of satellite observations of the hiss waves properties, e.g. from missions such as CRRES (e.g. Meredith et al. 2009), the Van Allen Probes (e.g. Li et al. 2015), and combined missions (e.g. Meredith et al. 2018a) or directly from the evolving observations of the ambient properties for a particular event (e.g. Ripoll et al. 2016b, 2017). The latter method is called the event-driven approach (e.g. Thorne et al. 2013) and is the focus of this article. It consists in feeding a quasi-linear Fokker-Planck model (here, we use the CEVA code developed originally by Réveillé (1997)) with in-situ measurements of wave properties and the plasma density observations made by the Van Allen Probes in order to produce pitch angle diffusion coefficients, $D_{\alpha\alpha}(t)$, at a high temporal resolution. The high temporal resolution comes from refreshing the coefficient values from the temporally updated parameters, with this new evaluation made at best at every pass of the satellite and properties assumed as constant between two passes. Results of Watt et al. (2021) have shown that updating the diffusion coefficients at a time rate of less than 9 hours (representing one Van Allen Probes orbit) was producing the best accuracy. In return, a computational step requires massively parallel computations in order to calculate bounced-averaged pitch angle diffusion coefficients at each satellite pass time, t, and location, L, i.e. $D_{\alpha\alpha}\left(t, L, E, \alpha\right) = D_{\alpha\alpha}\left(w_i(t, L), n_e(t, L), E, \alpha\right)$, with the locally measured wave properties denoted here as $w_i(t, L)$ for $i = 1...5$, and the plasma density, $n_e(t, L)$, for any electron at time t, L-shell L, of energy E, and equatorial pitch angle α. The index $i = 1...5$ includes the four main wave properties, which determine the distribution of a given wave both in frequency and wave normal angle, i.e., the wave mean frequency, frequency width, wave normal angle and wave normal angle width. The fifth parameter is the wave power, with a quadratic dependence of the diffusion coefficient on wave power. General and technical explanation about the computation of the diffusion coefficients are given in

the second section of this article. For further details of this method the reader is referred to Ripoll et al. (2017, 2019, 2020b).

Once diffusion coefficients are computed for a given event, one can repeat the procedure for many events of the same kind (here applied to high speed stream storms) and derive statistical event-driven diffusion coefficients $\tilde{D}_{\alpha\alpha}(w_i, n_e, L, E, \alpha)$, with $\tilde{D}$ denoting, for instance, a temporal average. For comparison, the classic statistical approach, for which the mean of the properties is used, produces instead $D_{\alpha\alpha}(\tilde{w}_i, \tilde{n}_e, L, E, \alpha)$. In statistical methods, a binning on the geomagnetic conditions (using the Kp geomagnetic index (Sicard-Piet et al., 2014; Horne et al., 2013) or sometimes the AE index (Meredith et al., 2018b)) is commonly introduced in order to reflect at least partially the dynamics of the wave-particle interaction. Conversely, our method allows to keep the non-linearity of the functional form of the diffusion coefficients and the coupling between all parameters since we compute means of diffusion coefficients $\tilde{D}_{\alpha\alpha}$ rather than diffusion coefficients of mean properties. We believe this is required to capture accurately the variability and complexity of atmospheric electron precipitation, which is crucial for studying the Earth's ionosphere-magnetosphere coupling. Similarly to statistical methods, we will also re-introduce a binning with respect to the geomagnetic indices once we generate statistics of event-driven diffusion coefficient below, i.e. means of diffusion coefficients $\tilde{D}_{\alpha\alpha}$ per geomagnetic activity bin, with the use of machine learning techniques.

Machine-learning (ML) techniques have been used for different problems related to ionospheric physics, such as ionospheric scintillation (Linty et al., 2018; McGranaghan et al., 2018), the estimation of maps of total electron content (TEC) (Tulunay et al., 2006; Sun et al., 2017; Cesaroni et al., 2020), the modeling of the foF2 parameter (which is the highest frequency that reflects from the ionospheric F2-layer) (Oyeyemi et al., 2005), the generation of maps of the thermosphere density (Pérez et al., 2014), and the forecast of electron precipitation (McGranaghan et al., 2021).

For radiation belt physics, neural networks (NN) are among the most popular machine learning methods. NN have been used for geomagnetic indices prediction, such as Dst/SYM-H, K_p, AE, and AL (Gruet et al., 2018; Siciliano et al., 2021; Takalo and Timonen, 1997; Bala and Reiff, 2012) (see also review in (Liemohn et al., 2018)). Models of plasmaspheric density have been developed in Zhelavskaya et al. (2016, 2017, 2018) and Chu et al. (2017b,a) using NN in order to compensate the lack of density data in radiation belt Fokker-Planck simulations. For instance, Ma et al. (2018) computed pitch angle and energy diffusion coefficients using the NN-based density model of Chu et al. (2017b,a) in the dusk sector where density can be hard to infer and used them afterward in Fokker-Planck simulations. Malaspina et al. (2018) use the NN-plasmasphere model of Chu et al. (2017b) to quantify the importance of the density for parameterized maps of whistler-mode hiss waves, and Camporeale et al. (2019) provide estimates of the uncertainty for the predictions of that NN-plasmasphere model. Other neural network-based models of plasmaspheric density have been developed in Zhelavskaya et al. (2016, 2017, 2018) and then used in radiation belt Fokker-Planck simulations. For instance, Wang et al. (2020) have performed simulations using plasmapause positions inferred from a combination of empirical and Zhelavskaya's NN-based density model and showed the importance of the plasmapause positions on the dynamics of relativistic electrons. For a detailed review of machine learning methods applied to both ionospheric and magnetospheric problems, the reader is referred to the review in Camporeale (2019).

In this article, we will show that we can construct a ML model for a single storm based on assimilating the pitch-angle diffusion coefficient $D_{\alpha\alpha}(t, L, E, \alpha)$. Ideally, in order to extend that model to the prediction of any storm, we would need quantities that describe the electromagnetic waves and the plasma conditions for each ongoing storm, which does not exist in practice. Here, we derive the simplest possible global

event-driven model encompassed within a ML model and built on an existing large database of event-driven diffusion coefficients. This means that we have to do prediction-error experiments, trying to model pitch angle diffusion for storms for all their given variables, evaluate model errors with the reference data, and modify the type or the number of the used variables to improve the model at best. A similar problem was addressed in Zhelavskaya et al. (2016) for a different quantity: the prediction of the cold electron density, training multiple neural networks with different variables and producing different time-averages. Time averaging is also at stake when constructing a global model: the longer the averaging period, the more regularized the model. With a regularized model, the machine learning model is easier to obtain, but its predictive ability is degraded considering a sample event. Yet, regularization should also help in generalizing the model to out-of-sample events.

As a first step, we construct a specific-event model using data from one storm (i.e. March 1, 2013). In other words, we build a regression model for $D_{\alpha\alpha}\left(L, E, \alpha\right)$ in 3 dimensions. We compare the results of 8 machine learning methods, such as deep neural networks, functional approximation and tree-based models, and we use different sizes of training dataset to test each model.

As a second step, we construct a global event-driven model $D_{\alpha\alpha}\left(t, K_p, L, E, \alpha\right)$ with a deep neural network using data from the 32 high-speed streams (HSS) storms. For each storm, we extract the geomagnetic index K_p evolving in time during the 3 days of the main and recovery phases of the HSS storms (Turner et al., 2019). Time will be kept as a main parameter and serve to produce a superposed epoch analysis of diffusion during the 3 first days of the HSS storms. This is based on the recognition that each storm has a time history, considering, for instance, that two storms having the same geomagnetic activity index at the beginning of the storm, or at the end, can still give different pitch-angle diffusion coefficients (as the data show). The deep neural network is thus used to learn from a giant diffusion coefficient database and construct the first statistical event-driven model diffusion coefficient by whistler-mode hiss waves during HSS events, parameterized by both epoch time and Kp index. The machine learning model is thus used to replace averages and interpolations of the database elements, which one would perform usually by hand, by a numerical expression, which is afterward extremely easy to call for any epoch time, K_p index, location, energy and pitch angle, without notably altering the accuracy of the initial database. The article is organized as follows. After the introduction in section 1, we present in section 2, the dataset and the machine learning methods that are used and tested in this study. In section 3, we present our results first for all methods for the March 1, 2013, storm with regularized data and, then, for the global, i.e. statistical, even-driven model diffusion coefficient of HSS events made from a database of 32 HSS storms. In section 4, we discuss the global DNN pitch angle diffusion model and its use for exploration of the database. Conclusions are given in Section 5.

2 MATERIALS AND METHODS

2.1 Description of datasets

2.1.1 Pitch angle diffusion coefficients

The diffusion coefficient represents the diffusive effect of a given electromagnetic wave (defined by its wave properties) on an energetic electron (with energy E and pitch angle α) trapped on a magnetic field line at a L-shell L in a medium containing cold electrons of density n_e. Equations 2 to 8 of Lyons et al. (1972) define the diffusion coefficients as they are used here. A more synthetic and modern expression of the diffusion coefficients is available through Equations 8, 9 in Mourenas and Ripoll (2012) using the notations of Albert (2005). One can see that the coefficient directly and explicitly depends on: wave amplitude,

wave frequency distribution (defined by a mean frequency and a mean frequency width), a wave normal distribution (defined by a mean wave normal angle and mean wave-normal-angle) and plasma density. Diffusion coefficients are computed with the CEVA code originally developed by Réveillé et al. (2001). In this code, bounce averaged diffusion coefficients are computed following the method and equations of Lyons et al. (1972), which account for a sum over all harmonics $(-n..., 0, ..., n)$, a wave normal integration, and bounce averaging between the mirror points. The limit of low frequency ($\omega_{med}/\omega_{ce} < 1$) and high-density ($\omega_{med}\omega_{ce}/\omega_{pe}^2 << 1$) are assumed in these computations. (See also Albert (1999) where this model is derived within these approximation and analyzed). Drift averaging is then performed in order to produce mean diffusion coefficients over the full electron drift. Verification by comparison with diffusion coefficients computed with the codes from the US AFRL and BAS (e.g. Albert 1994, 2008; Meredith et al. 2007) have been performed in Ripoll and Mourenas (2012). Validation studies of the CEVA code include Ripoll et al. 2016b, 2017, 2020b; Loridan et al. 2019; Millan et al. 2021.

Diffusion coefficients are evaluated from observed properties in a dynamic way so as to generate event-driven pitch angle diffusion coefficients. Event-driven diffusion coefficients are computed by temporal bins of 8 hours each day (3 bins a day). As time is frozen within a 8-hour bin and corresponds to roughly a full orbit of the Van Allen Probes, this allows to have frozen parameters for the whole L-shell range (from apogee to perigee of the probes) during each temporal bin. This is made to be able to solve the Fokker-Planck equation over the entire radiation belt regions through which trapped electrons are transported during storms and where they can interact with electromagnetic ambient waves (albeit the wave is present). An 8-hours temporal resolution also allows to account for short timescales causing non-equilibrium diffusion effects (i.e. solutions far from steady states) (e.g. (Ripoll et al., 2016a; Watt et al., 2021; Millan et al., 2021)). This means that we evaluate the diffusion coefficients with new properties each 8 hours during the few days the storm lasts. We use Van Allen Probes observations of wave amplitude, mean frequency, mean frequency width, mean wave normal angle, mean wave-normal-angle and plasma density so that all parameters are data-driven. Each one of these ambient properties changes with time and L-shells as the satellite observes a new value at each pass. In between two passes, we assume conditions are stable enough so that we can keep all parameters constant. This assumption is forced by the lack of available satellite data at higher rates. Eventually, the diffusion coefficients are specific to particular chosen events and qualified as 'event-driven' or 'event-specific'.

All the wave properties, which were listed above as $w_i(t, L)$ for $i = 1...5$, have been extracted from data of whistler-mode hiss waves (0.05 to 2kHz; e.g., (Santolík et al., 2001)). These primitive data are taken from measurements by the Electric and Magnetic Field Instrument Suite and Integrated Science (EMFISIS) Waves instrument aboard the Van Allen Probes (Kletzing et al., 2013). As we do, a Magnetic Local Time (MLT) dependence of the wave amplitude (i.e. the square root of the power) is taken into account by rescaling the locally observed wave amplitude by the MLT-dependence derived statistically from 4 years of Van Allen Probes data by Spasojevic et al. (2015). The latter approximation is required to account for the great variability of the wave amplitude with MLT (since measurements at all MLTs do not exist) but may introduce temporal inaccuracies due to the use of a statistical model. The MLT rescaling produces diffusion coefficients that apply over the full azimuthal drift of the electron. Similarly, dependence of the diffusion coefficient with the cold electron plasma density ($n_e(t, L)$) is accounted for by using either the density deduced from the upper hybrid line measured by EMFISIS (Kurth et al., 2015) or the density computed from spacecraft charging (Thaller et al., 2015) measured by the Electric Field Wave instrument (EFW) (Wygant et al., 2013) aboard the Van Allen Probes. We note that the wave properties are taken from past measured events and that they are unknown for future events so that any model of diffusion coefficients cannot be made with the wave properties set as mathematical variable. Wave properties remain

Name	# of Storm	time	L	E	α	# data	Comment
DS1	32	9	43	60	256	1.90E8	Raw data
DS2_L	1		37	5	60	1E4	Storm of March 2013, from DS1
DS2	32	9	4	60	256	1.8E7	Filtered in L, from DS1
DS3_M13	1		4	60	256	6.1E4	Storm of March 2013, from DS2
DS3_AVG	avg	9	4	60	256	2.3E6	Averaged (from DS2) global data

Table 1. List and properties of the various datasets in use.

mandatory parameters that one can either take from direct measurements as here or from statistical models (e.g. (Horne et al., 2013; Sicard-Piet et al., 2014; Spasojevic et al., 2015; Li et al., 2015; Ma et al., 2018; Wang et al., 2020; Cervantes et al., 2020)). Prediction can then be made from postulating a temporal series of one (or more) geomagnetic index for a given period of time or a known type of event.

Once the diffusion coefficients have been generated from all the primitive ambient properties, they only remain dependent on time t, L-shell L, energy, E, and equatorial pitch angle, α. The original spatial grid of the diffusion coefficients, $D_{\alpha\alpha}\left(t, L, E, \alpha\right)$, is composed of 43 uniformly distributed bins in L-shell, from $L = 1.3$ to $L = 5.5$. The energy grid is composed of 60 logarithmically distributed bins from $E = 50$ keV to $E = 6$ MeV. The pitch angle grid is composed of 256 uniformly distributed pitch angle, from the loss cone pitch angle to 90 degrees. This leads to 660480 values per time of interest.

Due to the large variability of the ambient properties, geomagnetic conditions, and position, the values of interest of the pitch-angle diffusion coefficient spread over many decades (from 10^{-19} to 10^{-4} s^{-1}) so that all our machine learning models will output the logarithm of the diffusion coefficient. However, all averages will be made directly on the pitch-angle diffusion coefficient, since averaging instead its logarithm would have weighted excessively the lowest diffusion coefficients and biased them.

During the storm evolution, some of the highest L-shells are located outside the plasmasphere where hiss waves are absent, which produces at best a null (when there are traces of the wave in some denser detached regions) or undefined diffusion coefficients (when the absence of the wave makes the main parameters missing). In this case, the coefficients need to be kept as a null pitch-angle diffusion coefficient in the database and in the statistics. If they were removed of the data, it would result in the rare events in which the wave are presents wrongly dominating the statistics.

2.1.2 Original full dataset

In this study, we consider either 1 or 32 storms, 1 or 9 time intervals, 43 positions, 60 energies, 256 pitch angles. This corresponds to 190 million data points, which we call the full dataset, DS1 in Table 1. This original set is too large for the herein regression in dimension 3 (i.e. L,E,α) or 5 (i.e. t, geomagnetic index, L,E,α) and the first task is a strategy to reduce the amount of data.

In this article, we first restrain the dataset by choosing values of L at a few discrete points $L = 2,\ 3,\ 4$ and 5, which gives around 18 millions data. Five L-shells are enough to be representative of the general behavior of the diffusion coefficients, i.e. the spread of the cyclotron component over pitch angle, in order to first focus on the reduction in (E,α) at fixed L. This dataset is called DS2, see Table 1. The reduction method in (E,α) is then directly extended to a finer grid in L in the case of the 32 storms global model (cf. section 2.1.4).

2.1.3 Dataset for the storm of March 2013

The dynamics of the electron radiation belts during the month of March 2013 have been subject to much attention (e.g., Baker et al. 2014; Li et al. 2014; Reeves et al. 2016; Ripoll et al. 2016b, 2017, 2019). The storm of March 1, 2013, is associated with a high-speed solar wind stream that created strong erosion of the plasmasphere and resulted in outer belt flux dropout events. The storm was followed by enhancements of relativistic electrons in the slot region and outer belt during the three days. An extended period of quiet solar wind conditions persisted then for the 11 next days, with the plasmasphere expanding outward to $L \sim 5.5$. For this event, Ripoll et al. (2016b) showed the electron depletion in both the slot region and the outer belt was caused by pitch angle scattering from whistler mode hiss waves. Ripoll et al. (2019) extended the demonstration to a global analysis of the 3D (L, E, α) structure of the radiation belts during the quiet times from 4 to 15 March and compared the output of event-driven Fokker-Planck simulations to pitch angle-resolved Van Allen Probes flux observations with good agreement.

In this section, we focus on the specific storm of March 1, 2013, and we use the event-driven diffusion coefficients database that was generated for the studies of Ripoll et al. (2019). Specific parameters of the diffusion coefficients are given there and not recalled here. These coefficients use the local wave and data parameters and as such can contain the noise and the variability of the measurements. But since the expression of the diffusion coefficients is made of the combination of tractable mathematical expressions, with some oscillating Bessel functions, and a series of summation (over the harmonics) and integration (over both frequency and wave normal angle) (e.g. Albert (1999)), the database ends up being quite smooth and not too noisy. This will be a key property of the data for choosing or developing an adapted machine learning method. In addition, the diffusion coefficients are also time-averaged from March 1 to March 5 in order to provide a single diffusion coefficient defined for L-shell L, energy E, pitch angle α. This time-averaging made over 5 days (representing 15 temporal bins of 8-hours averaged together) produces smoothed data, i.e. a regularized dataset, which may otherwise be more variable over time and less smooth (e.g. Figure 5 in Ripoll et al. (2017)). As we average, we mix different geomagnetic conditions and create a mean diffusion coefficient for that 5-day event. The time-averaging is only done in this section and will not be done in the HSS section in which we will keep time as another variable. Absence of noise and regularized data make our problem specific. On the contrary, in general, data have uncertainties coming either from our partial knowledge of the variables, or from data variability. In our case, we can have experimental and simulation uncertainties. In such cases, machine learning models have to avoid over-fitting, by not being too close to the data during training. In this article, regularization of data was such that over-fitting was not an issue.

For this storm, we use 4 positions, 60 energies, 256 pitch-angles, i.e. 61440 data points for $(L, E, \alpha, D_{\alpha\alpha})$ listed as DS3_M13 in Table 1. We extract a subset of DS3_M13 that is composed of 84 pitch angles and 60 energies bins, thus 20 160 data points, listed as TRAIN_M13 in Table 2. This dataset is used for training and calibrating the internal parameters of the various machine learning models using cross-validation.

To evaluate the ability of the machine learning models that we trained on the TRAIN_M13 dataset, to generalize on new data, we consider 2 test datasets, see Table 2. The first dataset TEST_M13_L contains more values in the L input. The model was trained with 4 L-values ($L = 2,\ 3,\ 4,\ 5$), and here we have 37 values from $L = 2$ to $L = 5$: thus we test the interpolation between the discretization used during the training in the case of a very low resolution. The other test dataset (TEST_M13) has full resolution in angles and energies, but the same resolution in L. The test datasets have no intersection with the training dataset. We have also excluded all extrapolation points (with an exception for K_p in section 3.2.3), signifying that we bound the test datasets with the bounds of the corresponding training datasets, when evaluating errors.

Name	Obtained from	How	# data	Comment
TRAIN_M13	DS3_M13	84 chosen α $8 \leq \alpha \leq 89$	20 160	all models trained
TRAIN_AVG	DS3_AVG	shuffled sampling	230 000	DNN trained (only)
TEST_M13_L	DS2_L	shuffled sampling, substraction of TRAIN_M13	5000	High resolution in L
TEST_M13	DS3_M13	by substracting TRAIN_M13	40 000	Test (L, E, α)
TEST_AVG	DS3_AVG	shuffled sampling, substraction of TRAIN_AVG	230 000	Test (K_p, t, L, E, α)

Table 2. The datasets used for training (2 first rows) and testing (3 last rows). Test data are obtained by substracting the training and validation datasets from the data, and also all points that are outside the bounds of these training and validation datasets, so as to avoid extrapolation in the test.

2.1.4 Dataset for the 32 HSS storms

In this section, we extend massively the previous problem from 1 storm to 32 storms. We choose storms all among the same family of storms called high-speed streams (HSS) so that we can compare them together, characterize the differences, and compute relevant statistics. By doing so, we try to optimize our chances to address similar physical processes and their spatio-temporal timescales. These 32 HSS were each identified in Turner et al. (2019) between September 2012 and December 2016 listed in Table 3). Each storm is observed at various MLT positions, changing with the Probes orbit. When Van Allen Probe B is at its apogee, the corresponding MLT is reported in the right column of Table 3. This MLT corresponds roughly to the most observed MLTs from L above ~ 4 up to $L \sim 6$. The 32 storms are such that we have 10 events observed from the night side (MLT=21-3), 11 from the dusk side (MLT=15-21), 4 from the day side (MLT=9-15), and 7 from the dawn side (MLT=3-9). Some of the differences we will found may be attributed to MLT variations, though keeping in mind that the statistical MLT-rescaling of the wave power makes the coefficients valid and comparable over all MLTs.

For each observed storm, we extract wave and plasma data from the Van Allen Probes during 3 days, every 8 hours, which gives 9 intervals of 8 hours. The timescale of 3 days is representative of the HSS main and recovery phases (Turner et al. (2019)). The measurements are used as inputs in the simulations of the quasi-linear pitch-angle diffusion coefficients (Ripoll et al. (2019)) outputted at this rate, producing the full database DS1.

For each storm and for a given time bin, we have a discretized grid (L, E, α) of the diffusion coefficient. For each temporal bin, we store the K_p index (itself averaged over the 8 hour bin duration). The Kp-index is the global geomagnetic activity index that is based on 3-hour measurements from ground-based magnetometers around the world. The Kp-index ranges from 0 (very little geomagnetic activity) to 9 (extreme geomagnetic storms). The Kp index is largely used in the radiation belt models as a main parameter of wave models driving radiation belt simulations (e.g. Cervantes et al. (2020); Sicard-Piet et al. (2014); Wang et al. (2020)). Here, it works as a measure of the storm strength at a given time. We define averages per K_p index and regroup the diffusion coefficients per K_p. The K_p index then becomes the 5th variable, which was first meant to replace the time variable, as any K_p average model, but we will explain later that time was nevertheless kept. As such, we have 18 millions of data points in $(t, K_p, L, E, \alpha, D_{\alpha\alpha})$, which gives data set DS2.

We build a first set of averaged diffusion coefficients by considering all the 32 storms, each defined at 9 temporal bins, which now define 9 epoch times. For a given temporal bin $j = 1..9$, for a given $K_p = 0...6$, we average $D_{\alpha\alpha}(L, E, \alpha)$ over all the storms. We obtained this way 2 300 000 data points, listed as

Event #	Minimum Date/Time	Min. SYM-H	MLT
1	2013-01-26/22:19:00.000	-6.2e+01	2.9
2	2013-04-24/18:11:00.000	-5.2e+01	23.1
3	2013-08-05/02:20:00.000	-5.6e+01	15.5
4	2013-08-16/04:29:00.000	-5.4e+01	15.1
5	2013-08-27/21:43:00.000	-6.4e+01	18.8
6	2013-10-15/03:18:00.000	-5.2e+01	17.2
7	2013-12-08/08:30:00.000	-7.2e+01	15.2
8	2014-02-23/22:48:00.000	-6.3e+01	12
9	2014-08-27/18:18:00.000	-9.0e+01	5.5
10	2014-10-14/18:38:00.000	-5.2e+01	3.7
11	2014-10-20/17:10:00.000	-5.7e+01	3.5
12	2014-11-16/07:24:00.000	-5.1e+01	2.5
13	2015-02-17/23:55:00.000	-7.0e+01	23.3
14	2015-02-24/03:36:00.000	-7.6e+01	23
15	2015-04-16/23:29:00.000	-8.8e+01	21.1
16	2015-05-13/06:59:00.000	-9.8e+01	20
17	2015-05-19/02:55:00.000	-6.4e+01	19.7
18	2015-06-08/07:45:00.000	-1.05e+02	18.9
19	2015-07-05/04:52:00.000	-5.8e+01	17.8
20	2015-07-23/07:28:00.000	-8.3e+01	17.1
21	2015-08-23/08:34:00.000	-6.2e+01	15.8
22	2015-10-04/07:33:00.000	-5.2e+01	14.3
23	2015-12-14/19:04:00.000	-6.0e+01	12
24	2016-02-18/00:28:00.000	-6.0e+01	9.5
25	2016-03-16/23:41:00.000	-6.9e+01	8.4
26	2016-04-13/01:09:00.000	-7.0e+01	7.3
27	2016-05-08/08:15:00.000	-1.05e+02	6.3
28	2016-06-06/06:47:00.000	-5.5e+01	5.3
29	2016-08-23/21:13:00.000	-8.3e+01	2.8
30	2016-10-25/22:57:00.000	-8.1e+01	0.4
31	2016-10-29/07:25:00.000	-7.8e+01	0.3
32	2016-11-25/06:38:00.000	-5.3e+01	23.2

Table 3. From left to right: number, Date and time, minimum Sym-H index (i.e. high resolution Dst index) and MLT of the apogee of probe B of the Van Allen Probes for each of the 32 high speed streams between September 2012 and December 2016 of this study (reported from the selection of the HSS events of Turner et al. (2019)).

DS3_AVG in Table 1. The model is defined for (t, K_p, L, E, α). Averages are made at fixed K_p for each t_j. (If we were averaging without binning by the K_p index, we would produce a superposed epoch model of diffusion coefficients). Here, the approach produces a superposed epoch model of the diffusion coefficient, further binned by K_p. Such an approach allows the diffusion coefficients to evolve in time, keeping within its origins ambient properties that are consistent with each other, always keeping the coupling between the electron plasma density and all wave properties. This approach is different from making a superposed epoch model of the wave properties of HSS and computing afterwards a single diffusion coefficients from them. The latter approach has low numerical cost but neglects correlations between all the properties of the ambient domain and, therefore, introduces some error (e.g. Ripoll et al. (2020b)). From a machine learning perspective, the Kp averaging helps producing smoothed data, acting as a regularization of the solution that makes the solution less fluctuating, i.e. less noisy from a ML-perspective, similarly to the temporally-averaged data of the March 2013 storm (as discussed in section 2.1.3). From DS3_AVG, we train on 10% of the data, listed as TRAIN_AVG in Table 2. All datasets are described in Table 1, training and validation datasets in Table 2 (2 first rows), and test datasets in Table 2 (3 last rows).

2.2 Machine learning methods

In this section we briefly describe the several statistical and machine learning methods that we used to build the various models of this study. We considered methods based on local evaluation (k-nearest neighbors and kernel regression), tree-based methods (regression tree, bagging and random forest), neural networks and function approximations (Radial basis and splines). All are nonparametric so that we make no assumption about the distribution of the data. A detailed description of all these machine learning methods can be found in Hastie et al. (2009), and complementary informations about neural networks can be found in Géron (2017), Goodfellow et al. (2016).

2.2.1 k-nearest neighbors (KNN)

A key idea in many supervised machine learning methods is to think that the targets associated to nearby inputs should be close to each other. Based on this idea, to predict the target of any new input data points, it is reasonable to look at the target values of their surrounding neighbors. This is the whole framework of k-nearest neighbors machine learning method which predicts the target of a new input data by averaging the target values of its k-nearest neighbors, measured using the Euclidean distance (see, for example, Fix and Hodges (1951); Altman (1992) and Hastie et al. (2009)). The number of nearest neighbors k is the key parameter and it is very crucial to tune it using cross-validation technique described in the following. On one hand, if k is too large, a large number of observations, among which not very representative ones, contribute to the prediction, resulting in too rough predictions. On the other hand, if k is too small, the prediction is made relying only on a small number of neighbors of the query point, resulting in high variance.

2.2.2 Kernel regression (KerReg)

The k-nearest neighbors procedure may be modified to obtain a smoother method, which gives more weight to the closest points and less to the furthest: instead of specifying a number of neighbors, the neighborhood is defined according to a distance notion, via a kernel function, that is a function $K : \mathbf{R}^d \to \mathbf{R}_+$, such that $K(x) = L(\|x\|)$, where $x \mapsto L(x)$ is nonincreasing. More specifically, a prediction $\hat{y}$ of a new data point x is obtained by setting :

$$\hat{y} = \frac{\sum_{i=1}^{n} K_h(X_i - x)Y_i}{\sum_{i=1}^{n} K_h(X_i - x)},$$

where the kernel K_h is defined by $K_h(x) = K(x/h)$, with h the bandwidth of the kernel, and (X_i, Y_i), $i = 1, 2, ..., n$, denotes the input-output training data. Here, a Gaussian kernel has been considered:

$$K(x) = \exp(-\|x\|^2/\sigma^2),$$

for some $\sigma > 0$. For more about the method see, for example, Nadaraya (1964) and Watson (1964).

2.2.3 Regression tree (Tree)

Another nonparametric model commonly used in regression problems is regression tree. It is an iterative partitioning algorithm aiming at each step to split the input space along the value of a chosen predictor and threshold, minimizing the target variance on both parts of the split (see Breiman et al. (1984)). Growing a tree is equivalent to partitioning the input space into smaller and smaller regions containing lesser and lesser points. The prediction of a new data point is the average target values of the points falling into the same region as the query point. Growing a single deep depth tree on the training data (small terminal nodes

or small region) will most likely lead to over-fitting. Moreover, a deep depth tree can be very sensitive (high variance) meaning that a small change in splitting the training data can result in a very different structure of the tree. It is then important to tune the depth of the tree, which is the key parameter. This may be done using cross validation technique.

2.2.4 Bagging (Bag)

The aim of this method is to reduce the variance of regression trees by introducing bootstrap samples from the training data. A regression tree is grown on each bootstrap sample, and the final prediction is the average of the predictions of all the trees (see Breiman (1996)). This method is shown to be significantly more accurate in generalization capability. The parameters of the method are the number and the depth of the trees to be constructed on the bootstrap samples.

2.2.5 Random Forest (RF)

As each tree in Bagging method is constructed using a bootstrap sample of the training data, the constructed trees are likely to be quite correlated. Random forests have been proposed to enhance reduction of the variance. They aim at producing uncorrelated trees by randomly selecting only a subset of features at each split in the process of growing the trees. In regression problems, the size of the set of features to be randomly selected at each split is usually taken around $\sqrt{p}$, where p is the total number of features (see, for instance, Ho (1995) and Breiman (2001)). In addition to the number of selected features, the parameters of the method are the depth and the number of trees.

2.2.6 Neural Networks (DNN)

We use feed-forward neural networks as a regression model. A neuron is the composition of a nonlinear function (here we use $Relu(x) = max(0, x)$) and a linear function. All inputs enter the $N1$ neurons of the first layer. Then each neuron gives an output, and each output connects to the $N2$ neurons of the second layer. We do the same for all the layers (the number of such layers is the depth of the network), and we end with a layer of one neuron (because we have one output, the pitch-angle diffusion coefficient), which has no nonlinear function. It has been shown (Cybenko, 1989) that any reasonable function may be approximated by one layer of neurons, but the practice has showed that it is better to go deep, which means to use a lot of layers (which entails a lot of composition of nonlinear functions, that is to say a lot of interactions between the inputs).

The coefficients of the linear functions of all the neurons are tuned by an optimization algorithm. This phase is called the training. We use a variant of the stochastic gradient descent method (the Adam optimizer) to minimize the mean square error between data and predictions.

Neural networks are accurate for regression problems, and extend well to huge dataset, or to high dimension problems. One difficulty is that such a model involves a lot of hyperparameters, and many combinations of these hyperparameters may give low accuracy results. For example, we have to choose the architecture (number of layers and neurons per layer), the initialization of the linear coefficients, the optimization algorithm, the number of epochs (iterations of the algorithm) and batches (splitting of the data to calculate gradients in the stochastic gradient descent). In order to optimize this choices, an original specificity of our DNN model is to use a data-driven method for selecting all these hyperparameters (Humbird et al., 2019; Kluth et al., 2020). It uses random forest methods (which has a few hyperparameters, see section 2.2.5) and a mapping between the obtained trees and the architecture of an ensemble of neural networks. We obtain this way accurate neural networks with only 2 hyperparameters, the depth and the

number of trees. When we obtain this accurate network, we may search for higher accuracy by playing with other hyperparameters that were fixed in the first step.

2.2.7 Thin plate spline (Spline)

Thin plate splines, introduced by (Duchon, 1977), may be seen as an extension of cubic smoothing splines to the multivariate case (Green and Silverman, 1994). In the one-dimensional case, cubic smoothing splines are used to construct new points within the boundaries of a set of observations. They are fitted using a penalized least squares criterion, with the penalty based on the second derivative. The interpolation function consists of several piecewise cubic polynomials. Fitting low-degree polynomials to small subsets of values instead of fitting a single high-degree polynomial to all data allows to avoids the Runge phenomenon, that is oscillation between points occurring with high-degree polynomials. Cubic smoothing splines are widely used since they are easy to implement and the resulting curve seems very smooth. More specifically, if we observe data $(X_1, Y_1), \ldots, (X_n, Y_n)$, the quantity to be minimized is defined by

$$\|\mathbf{Y} - \mathbf{f}\|^2 + \lambda \int (f''(t))^2 dt,$$

where $\mathbf{Y}$ is the vector of observed outputs $Y_1, \ldots, Y_n$ and $\mathbf{f} = (f(X_1), \ldots, f(X_n))$. In the general case, the main part of the criterion remains the same, but the shape of the penalty is far more involved, based on several partial derivatives. Thin plate splines are given as functions $\mathbf{f}$ minimizing

$$\|\mathbf{Y} - \mathbf{f}\|^2 + \lambda \text{pen}(\mathbf{f}),$$

where

$$\text{pen}(\mathbf{f}) = \int_{\mathbf{R}^d} \sum_{\nu_1 + \cdots + \nu_d = m} \frac{m!}{\nu_1! \ldots \nu_d!} \left(\frac{\partial^m \mathbf{f}}{\partial u_1^{\nu_1} \ldots \partial u_d^{\nu_d}} \right)^2 du,$$

and the factor λ drives the weight on the penalty. Here, m is such that $2m - d > 0$, and the ν_i's are nonnegative integers such that $\sum_{i=1}^{d} \nu_i = m$.

2.2.8 Radial Basis Function Interpolation (RBF)

A Radial Basis Function (RBF) is a function that depends only on the distance between the input and a predetermined fixed point, called a node. We can use RBF as a basis for an interpolator in the form:

$$f(x) = \sum_{i=1}^{N} h_i \phi_i(x), \tag{1}$$

where N is the number of nodes, h_i are unknown coefficients, and $\phi_i(x) = ||x - x_i||$, with x_i the coordinates of the i-th node. Here, we use all the points in the training set as nodes. The training consists in finding the values of the coefficients h_i by imposing that the interpolant passes exactly through the targets in the training set, that is $f(x_i) = Y(x_i)$. This amounts to solve the linear system $\mathbf{X}\overrightarrow{h} = \overrightarrow{Y}$ for the vector $\overrightarrow{h} = (h_1, \ldots, h_N)^T$, where $\mathbf{X}$ is the $N \times N$ symmetric matrix containing all the distances between nodes. Once we have the coefficients h_i, the targets in new data points can be evaluated directly by using the interpolator in Eq. (1). Even though the RBF could be generalized by introducing hyper-parameters (called in this context shape parameters), for instance defining $\phi_i(x) = ||x - x_i|| + c_i$, in this work we have not investigated more general choices of RBF and used only the form in Eq. (1).

2.2.9 Cross-validation

Each method depends on some key smoothness parameters (usually called hyperparameters) that need to be tuned properly to get a good performance. This is done via cross-validation. K-fold cross-validation consists of breaking down the training data into K folds $\{\mathrm{F}_k : k = 1, 2, ..., K\}$, and for a given candidate parameter, the corresponding model is constructed using as training set the $K-1$ folds where the remaining fold is treated as a validation dataset. Thus, for a given value of parameter β, the corresponding model f is trained K times (K different combination of $K-1$ folds choosing from the total K folds). We then measure the performance of f at the choice of parameter β using the cross-validation error defined by

$$\mathrm{CVE}(\beta) = \frac{1}{K}\sum_{k=1}^{K}\sum_{x_i\in\mathrm{F}_k}(f(x_i)-y_i)^2$$

In the particular case where each data subset only contains one single observation, the method is called leave-one-out cross validation.

Roughly speaking, this provides the average performance of f associated with the parameter β on K different unseen folds of the training data. The parameter $\hat{\beta}$ minimizing this cross-validation error would be a suitable one to be used as a global parameter in predicting the real testing dataset.

For k-nearest neighbors, kernel regression, regression tree, bagging and random forest, a 10-fold cross validation was used. For thin plate splines, the penalty coefficient is estimated through generalized cross validation, which may be regarded as an approximation to leave-one-out. For the neural networks, the training data set was randomly cut in 3 parts : 80% for the training, 10% for checking over-fitting during the training, and 10% for selecting the final network. After that hyperparameters selection, all results presented in this article are obtained on huge separated test dataset, as showed in Table 2.

2.2.10 Complexity of the training and computational time

Training phases are very different between all methods: for KNN there is only a search over the existing space of data. In tree-based methods the training corresponds to the construction of the trees. In DNN the training corresponds to the search for the weighting factors in the interconnections. All training phases agreed in the choice of the hyperparameters: as data have no uncertainties, and are somehow regularized, our methods have to fit to the training data. This means for tree-based methods to grow deep trees (one point in the final node), to be very localized for the k-nearest neighbors method ($K = 2$) and kernel regression, and to go deep with neural networks, with many epochs. Ensemble methods do not need to be pushed too far: for tree-based methods, we used 100 trees, and for neural networks, we averaged the outputs of around 5 networks. Moreover, thin plate splines are specifically dedicated to interpolation.

Even if the methods depend on the choice of hyperparameters, we can still say that the cpu-cost of training is about a minute for both regression tree and k-nearest neighbours, about 10 minutes for bagging and random forests, and 2 hours for neural networks, with each method using around 20000 data. Predictions are fast for all methods, meaning they take a few seconds maximum for 60000 data.

3 RESULTS

3.1 Results for the storm of March 2013

The numerical results reported in the following tables and figures present an analysis of the distribution of the errors: $e_i = y_i - \hat{y}_i$, for $i = 1, 2, ..., n$, for the different investigated methods, where $y_i = log(D_{\alpha\alpha})$, $\hat{y}_i$ is the prediction of the considered model, and n is the size of the considered test dataset. We train our models on the TRAIN_M13 data set, containing 20160 samples.

The TEST_M13 test data set is detailed in pitch-angles and energies but contains only 4 discrete L-shells values ($L = 2,\ 3,\ 4,\ 5$). The TEST_M13_L data set is however detailed in L and contains L-shell values regularly spaced from $L = 1.6$ to $L = 5.2$ by 0.1 step (37 values) and a few angles and energies values. These datasets are sampled on a grid and there is no uncertainty on the points. Hence, as already mentioned, all models are trained until reaching a small error value on the training data set. We first start by addressing the error with respect to the (E,α) resolved grids and then on the grid resolved in L-shell.

3.1.1 Variation with (E,α)

Results in table 4 show that the Spline, the RBF and the DNN models outperform with the lowest mean and maximal absolute error. We also observe that the Spline and the RBF have very low medians which show that they are very good on many samples, but have also many outliers, with big error. The DNN shows a median error close to the mean.

	Mean	Std	Q_1	Med	Q_3	Max
Tree	0.014	0.026	0.001	0.005	0.014	0.620
Bag	0.012	0.025	0.001	0.004	0.012	0.483
RF	0.012	0.025	0.001	0.004	0.012	0.448
KNN	0.010	0.017	0.002	0.006	0.011	0.420
KerReg	0.005	0.014	0.000	0.001	0.003	0.466
RBF	0.002	0.009	0.000	0.000	0.001	0.349
Spline	0.002	0.009	0.000	0.000	0.001	0.394
DNN	0.003	0.008	0.001	0.002	0.003	0.302

Table 4. Performances of all the methods trained on TRAIN_M13 and tested on TEST_M13. We consider the absolute error $|e_i|$ and report the mean error, standard deviation, first quartile, median error, third quartile, and maximal error.

The violin plots in the top panel of Figure 1 complement well Table 4 in showing that the underlying statistical distributions of the errors differ from one method to the other. DNN, Splines and RBF have the most concentrated distributions at low errors, especially both the Splines and RBF methods, with spline with a mode around the mean and RBF with a mode around the median. As seen on Table 4 with the maximal error, this hides more outliers with Splines and RBF than with DNN.

This results start to exhibit two main families of methods: on one hand the Tree family (Tree, Bag, RF) with KNN and KerReg and on the other the RBF, spline and DNN methods.

In order to get insights on the differences and similarities between the machine learning models, we now compute the correlation between the errors provided by couples of different models. The correlation errors are given in Figure 2 for TEST_M13, on the left. First, Figure 2 confirms the 3 methods (Tree, Bag, RF) fall into the same family, with high correlation (> 0.8). We will see in the next section that KNN and KerReg will join this same family but this is not obvious from the correlation errors of the left part of Figure 2. On

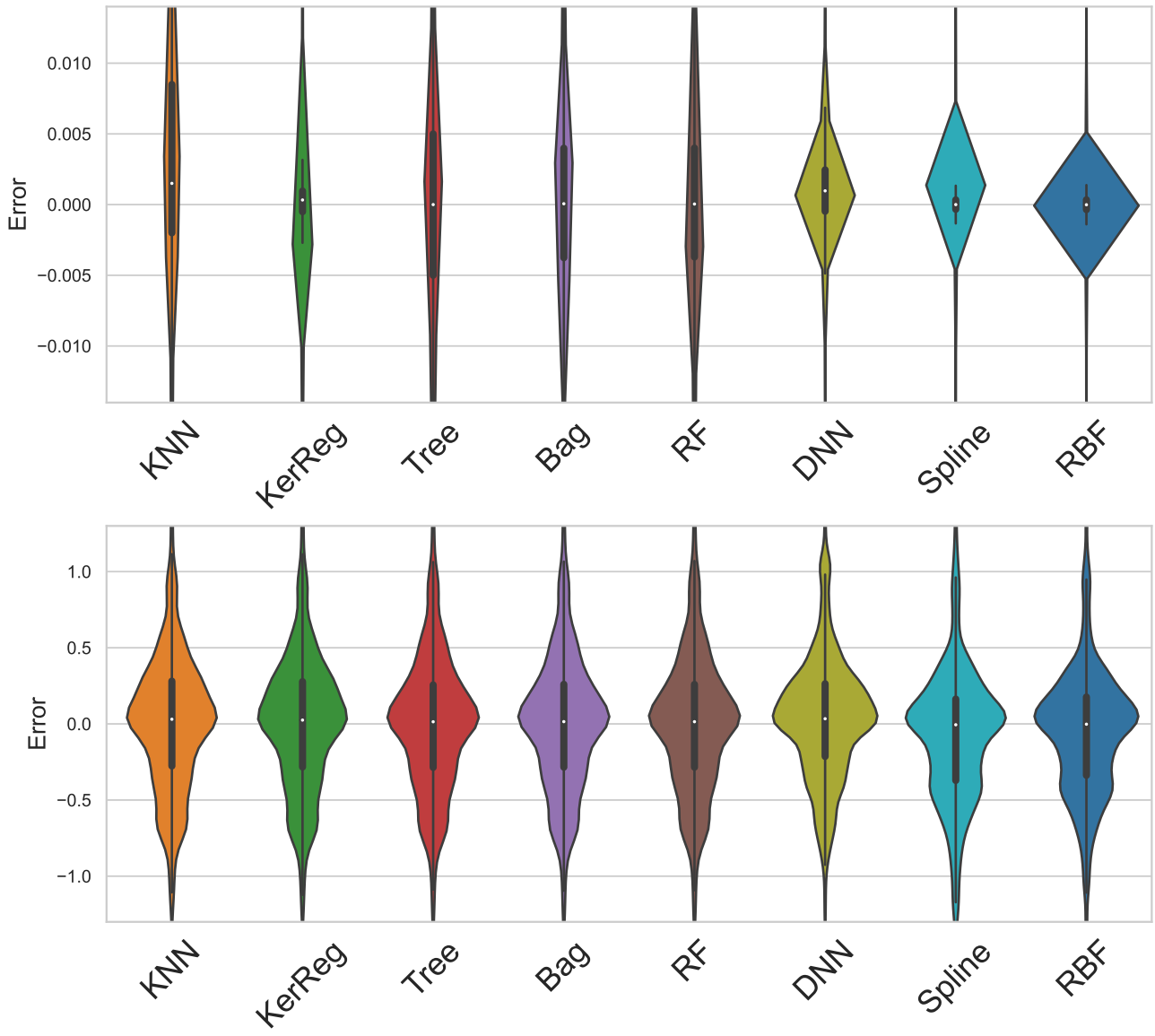

Figure 1. Violinplots of error $e_i = y_i - \hat{y}_i$ evaluated on (top) TEST_M13 and (bottom) TEST_M13_L of all the methods, trained on TRAIN_M13. For each ML-method, the outside envelop is the smoothed distribution of error, symmetric for visualization consideration, with a box-whiskers plot inside (median with a white circle, 1st and 3rd quartiles are represented by the border of the box).

the contrary, we see the specificity of the DNN method which errors does not correlate with any of the other method. The closest methods to DNN are the spline and RBF methods. Similarly, the spline and RBF methods correlate less with the forest tree method family.

Finally, this study was also conducted at low resolution with 13 energies and 14 pitch angle resolution, representing 728 data points (results not shown). Although some small changes of behavior either within or among the methods were visible, the conclusions were similar, for an admissible accuracy of the diffusion coefficients. Machine learning methods can thus be used to find an optimum between accuracy and resolution, reducing this way the high original cost of computation of the diffusion coefficients.

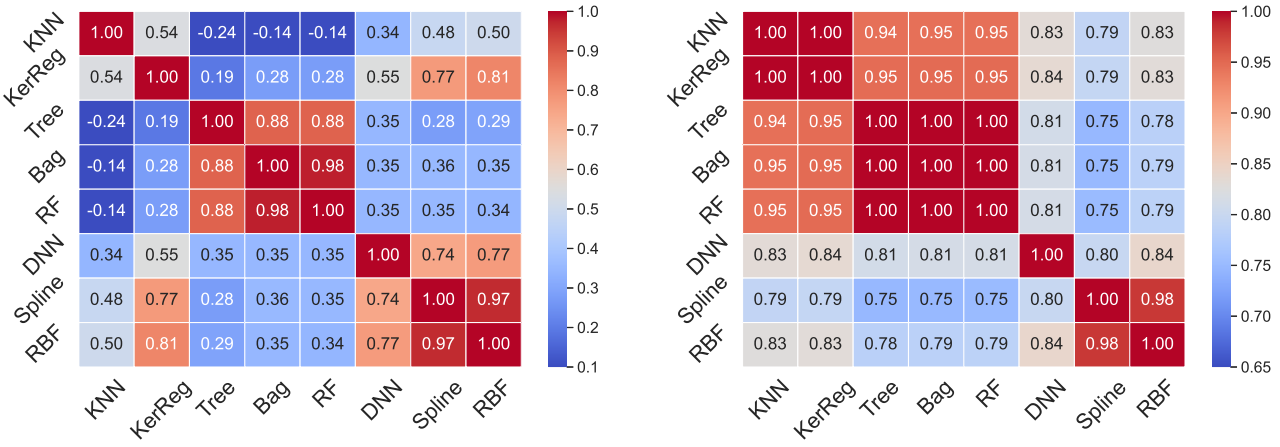

Figure 2. Correlations of error $e_i = y_i - \hat{y}_i$ evaluated on (left) TEST_M13 and (right) TEST_M13_L for all methods, trained on TRAIN_M13.

3.1.2 Variation with L-shell

In this section, we consider each method for its ability to capture the L-shell dependence. It should be noticed that all methods have been trained on a very crude L-shell resolution, containing 5 L-shells only, and that they are now tested against data fully resolved in L-shell. This test is therefore very challenging and only made to gain insight on the properties of the ML methods. If a full model in (L,E,α) had to be generated (cf. section 3.2), the approach would be to train on a higher L-shell resolution and not to interpolate a low resolution grid.

Tables 5 presents the main error global metrics, with errors much higher than in the previous section due to the initially low L-shell resolution. The mean error gradually decays from the Forest tree family to DNN (from top to bottom). However the median error remains more similar, still decaying from top to bottom. Best performances are always obtained from either the Spline, the RBF or the DNN method.

Violin plots of the distributions of errors have been generated (on the bottom of Figure 1). All the distributions are found very alike in their global shape, with only subtle differences. Some methods show two or even three modes which appear as peculiar oscillations on the edge of the distribution.

	Mean	Std	Q_1	Med	Q_3	Max
Tree	0.371	0.419	0.102	0.265	0.521	4.451
Bag	0.372	0.418	0.102	0.269	0.524	4.448
RF	0.372	0.418	0.102	0.267	0.525	4.459
KNN	0.364	0.378	0.108	0.279	0.520	4.374
KerReg	0.363	0.379	0.111	0.280	0.512	4.350
Spline	0.339	0.320	0.099	0.255	0.462	2.332
RBF	0.316	0.297	0.100	0.234	0.440	2.587
DNN	0.315	0.306	0.100	0.237	0.439	3.063

Table 5. Performances of all the methods trained on TRAIN_M13 and tested on TEST_M13_L. We consider the absolute error $|e_i|$ and report the mean error, standard deviation, first quartile, median error, third quartile, and maximal error.

Figure 2 (on the right) shows the error correlation among the couples of models for the test with respect to L-shell. The main families previously mentioned remain, this time with KNN and KerReg performing

similarly to the Forrest tree family. Based on these results and the one of the previous section, all 5 models (Tree, Bag, RF, KNN, KerReg) are regrouped into the same family.

We finalize this series of tests by Figure 3, in which we compare the forest tree family (represented by the RF method), the spline method, and the DNN method for a few selected (E,α) but resolved in L-shell. The training phase being done at $L = 2, 3, 4, 5$ (indicated by vertical bars), we see all models provide an exact answer at these points. Everything in between these points is modeled (orange line plots) and compared with the exact solution (blue crosses). The random forest model plotted in Figure 3 (left) uses constant approximation around the training points so that the approximation is made by step functions and is extremely crude. It is the same for tree-based methods, k-nearest neighbor and kernel regression (not shown). The spline method does much better in Figure 3 (center), but cannot approximate brutal variations, as for radial basis model (not shown). The DNN method in Figure 3 (right) seems to us the most capable for this difficult exercise, which confirms the global metrics of Table 5.

We conclude that without any prior assumption on a physical phenomenon and on the database, it is difficult to advise the use of a particular machine learning model. One main reason is the data used to train the model have a big influence on the model performance, which makes hard to generalize the capabilities of a given model. Here, we believe the different series of tests and comparisons are explicit enough to conclude that the DNN method is a good candidate to perform the rest of the study and to generate a more global model.

3.2 Results for the global model of pitch angle diffusion during HSS storms

In this section, we use the data from 32 storms in order to build a database of statistical event-driven diffusion coefficients that is embedded within a machine learning model for facilitating its use. The method relies on constructing first an averaged model and then using the deep neural network (DNN) previously selected to learn and output the solution of the averaged model. As in the previous sections, we will see the machine learning model does not entail any issue to interpolate and reproduce the averaged model. Questions arise more about the physical choices we make to build the averaged model (cf. discussion below and in section 4.1). Interestingly, the machine learning model was of great help for the various investigations we conducted. As the training step was quite fast (based on the knowledge acquired during the March 2013 storm study), we could test different ways of manipulating and averaging the data when iterating to choose how to best parametrize the statistical model. Another strength of the machine learning approach is the simplicity of performing comparisons with model since it delivers continuous maps of the solution with a simple numerical subroutine able to output a 5 to 6 dimensions solution. On the contrary, manipulating directly the database and use discrete points is very constraining. It can also be source of direct errors or interpretation errors when it is a given plotting software (e.g. Python subroutines) that carries intrinsic ad-hoc interpolation with integrated smoothing procedures.

3.2.1 Training the DNN global model

The data used to generate the global model is DS3_AVG described in Section 2.1.4 with $2.3e6$ data points. We then use TRAIN_AVG ($2.3e5$ data points), unless specified differently for training and validation, and TEST_AVG ($2.3e5$ data points) for test.

The training of the global model $D_{\alpha\alpha}(K_p, t, L, E, \alpha)$ was not harder than the model previously trained for the storm of March 2013 in section 3.1. The two more dimensions of the input space entailed a larger neural network. The bigger amount of data (from 20000 to 230000) caused a longer training. Generating the whole model took a few hours of computation on a standard computer. For comparison the simple

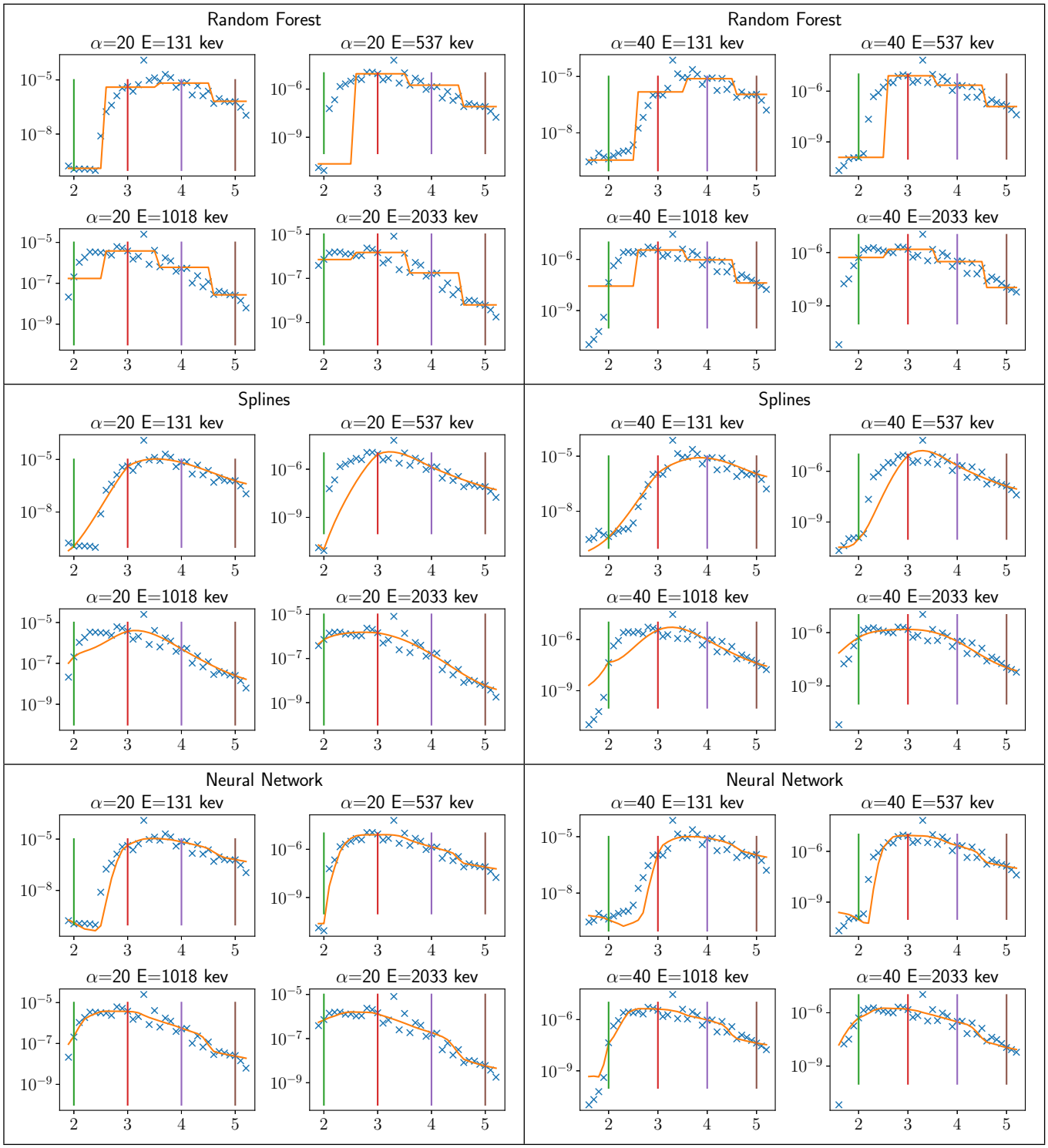

Figure 3. Machine learning pitch angle diffusion coefficients ($D_{\alpha\alpha}(L)$ in s^{-1} on the Y-axis) for the March 2013 storm plotted in orange color versus L-shell (X-axis) for various pitch-angles (20° and 40°) and energies (131, 537, 1018, 2033 Kev) computed from (top) random forest, (center) thin plate spline, (bottom) neural network. The blue dots are the reference original diffusion coefficients (points of TEST_M13_L which were not used in the training and testing phase). Vertical lines represents the location of the training data of TRAIN_M13 ($L = 2, 3, 4, 5$). These plots were made with methods that were trained on a subset of TRAIN_M13 : we used fewer points in energy and pitch-angles.

generation of a mean model, without Machine learning and performing only means throughout the whole database, took a few days on the same computer. Again for comparison, computing 19.3M diffusion coefficients for around 10-day event takes around 15600 hours spread over 1300 processors on a CEA massively parallel supercomputer (Ripoll et al., 2020b). This brings another advantage of machine learning methods to be able to manipulate simply and at low cost large database, with the possibiliy to operate on them basic statistical operations useful for the understanding of the database.

In Figure 4 (left), we represent the mean average error (MAE), the mean square error (MSE), and one minus the explained variance score (EVS) computed when the model is evaluated against the TEST_AVG test dataset (230 000 data not used during the training phase performed with the TRAIN_AVG dataset). Because the dataset contains little noise, we can train neural networks going deep, with depths of the network going from 6 to 11 hidden layers on the x-axis. We see an optimum of low values of the three metrics is found for a depth of 9.

The same three quantities are plotted in Figure 4 (center) using different sizes of training dataset (from 1% to 15% of the TRAIN_AVG dataset), with DNN of depth 8 or 9. From these results, we selected the neural network of depth 9 trained with 10% of the data. As over-fitting is not an issue here, we could reach better accuracy by taking a more important capacity for the model, or just by taking more epochs as discussed next. This is not obvious on Figure 4 as with a higher depth, error is growing (after depth 9), but it is possible to be more accurate by varying all hyper-parameters. However, we have also seen that the loss of accuracy due to the DNN model is less the issue than the loss of accuracy caused by an averaged statistical approach (cf. discussion in section 3.2.2).

Figure 4 (right) represents the loss function during the training and the validation phases of the model of depth 9 over 10% of the data. The loss function is the minimal MSE computed over all the data and evolving according to the epoch number, which represents the number of cycles the data are used in a training or validation step. After 900 epochs, we evaluate more often the loss function, because we stop at the best loss value obtained on the validation dataset.

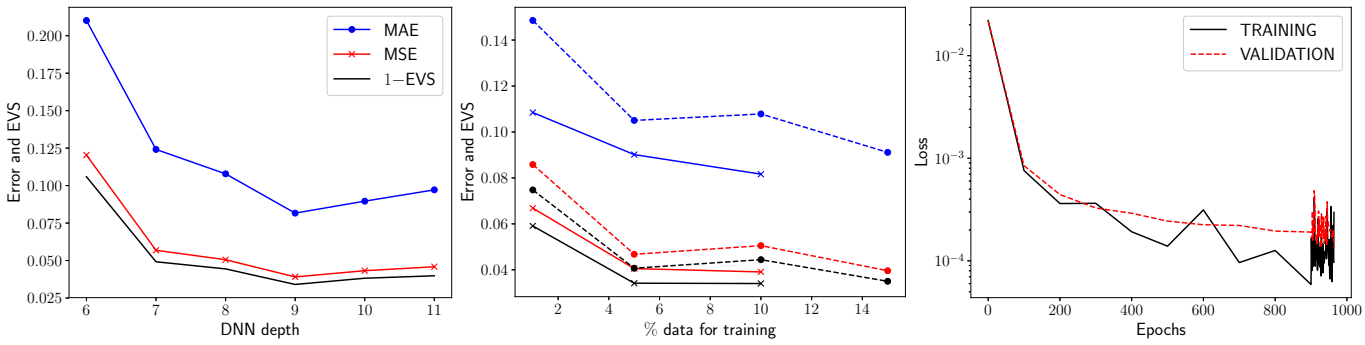

Figure 4. Errors calculated on the dataset $TEST\_AVG$ are plotted for different DNN models, with various depth on the left, and various sizes of training dataset on the middle. On the left, models are trained on 10% of the data, meaning around 230 000 data. On the middle, dot lines with circles are for a model of depth 8, and continuous lines with crosses for a model of depth 9. In blue, we plot the Mean Average Error, in red the Mean Square Error and in black one minus the Explained Variance Score. On the right, the loss function (MSE) is plot during the training, evaluated on the training dataset, and on the validation dataset. We see at the end that we make more often evaluations of these errors, and the training stops selecting the more accurate model in the last epochs. Note that this loss may not be compared with anything in this article, as it is given on scaled data.

3.2.2 Accuracy of the DNN global model

We present in Figure 5 the obtained deep neural network (DNN) global model of diffusion coefficients, which is plotted in green for two of the 32 selected storms. We choose for illustration storm 8 (3 top rows) as event-driven and average diffusion coefficients agree quite well and storm 5 because the opposite occurs. We also plot in red the average model, which represents what the DNN model (in green) has to reproduce. Each storm is decomposed in 9 times with its K_p index history (as shown in the bottom panel of Figure 5) and the DNN model is played for the (t,K_p(t)) sequence of this storm. Results are presented at L=3, 4, 5. We omit L=2 for the sake of brevity since diffusion is limited to high energy (see discussion in section 3.2.3 and Figure 6 top row).

As we can see for storm 8, the DNN model is very close to the average model, as it should, as soon as the intrinsic interpolation rules of the model have been learnt well. This is confirmed for storm 5 in Figure 5, which ends our demonstration that the restitution of the DNN model is accurate. Figures like Figure 5 (3 top rows) have been generated for each of the 32 HSS storms (not shown), which allows us to reach an individual view of each of them and confirm the accuracy of the DNN approach. This occurs at all L-shells used to derive the DNN model.

We now use the DNN mean model to analyze how mean coefficients behave compared with individual ones. An important physical question arising in space weather forecasting is the ability of an average model (e.g. from the DNN approach or directly from averaged data) to precisely predict the history of the diffusion during the storm. We thus compare in Figure 5 the DNN statistical model (in green) with the event-driven diffusion coefficients (blue cross). We find the average procedure captures quite well the global variations of the pitch-angle diffusion coefficient in general for storm 8 but fails by a significant factor at various (t,L,E,α). This way we start to enlighten the difference between an event-driven approach and a mean approach thanks to the machine learning interface. We see for instance a interesting strong departure at (L=3, E=0.3 MeV, α =60°, t= 1.6 days) for storm 8 between both the average models (green and red) and the event driven model (blue). Readers will understand in the next section (based on Figure 6 top, left) that α =60° falls right at the sharp edge between significant diffusion of the cyclotron harmonics and absence of diffusion for E=0.3 MeV electrons at L=3. Both average models capture thus (on average) significant diffusion while for storm 8 at t=1.6 day the diffusion is negligible, causing an error by more than 2 orders of magnitude. Note that all models agree for the time before (t=1.3) and after (t=2.). This is likely due to the particularity of the wave conditions at t=1.6. Conversely, Storm 5 (fourth to sixth rows of Figure 5) is an example of the opposite, with a storm for which the diffusion coefficient behavior (in blue) is opposite to the mean behavior (red and green). The error between the average model and the event-driven coefficient is often large, up to 2 orders of magnitude. We see the same feature as for storm 8 at L=3, E=0.3 MeV, α =60. Large errors at L=2 (not shown) for 1 MeV electrons are also likely due to the average model missing the particularity of a local increase of diffusion close to a strong gradient region. At L=5, we see the absence of the event-driven coefficients for that case, except for the point at the latest time, at t=3 day. This can be due to the plasmasphere that has not recovered up to L=5 during the first 2.6 days and the absence of hiss waves, to the absence of measurements for that event, or both. The average model returns low diffusion most of the time (below 10^{-6}), except for E=0.3 MeV and α=60.

3.2.3 Exploring the DNN global model

We now explore and discuss the main physical characteristics of the statistical mean model of pitch angle diffusion coefficients for HSS storms thanks to the DNN encapsulation.

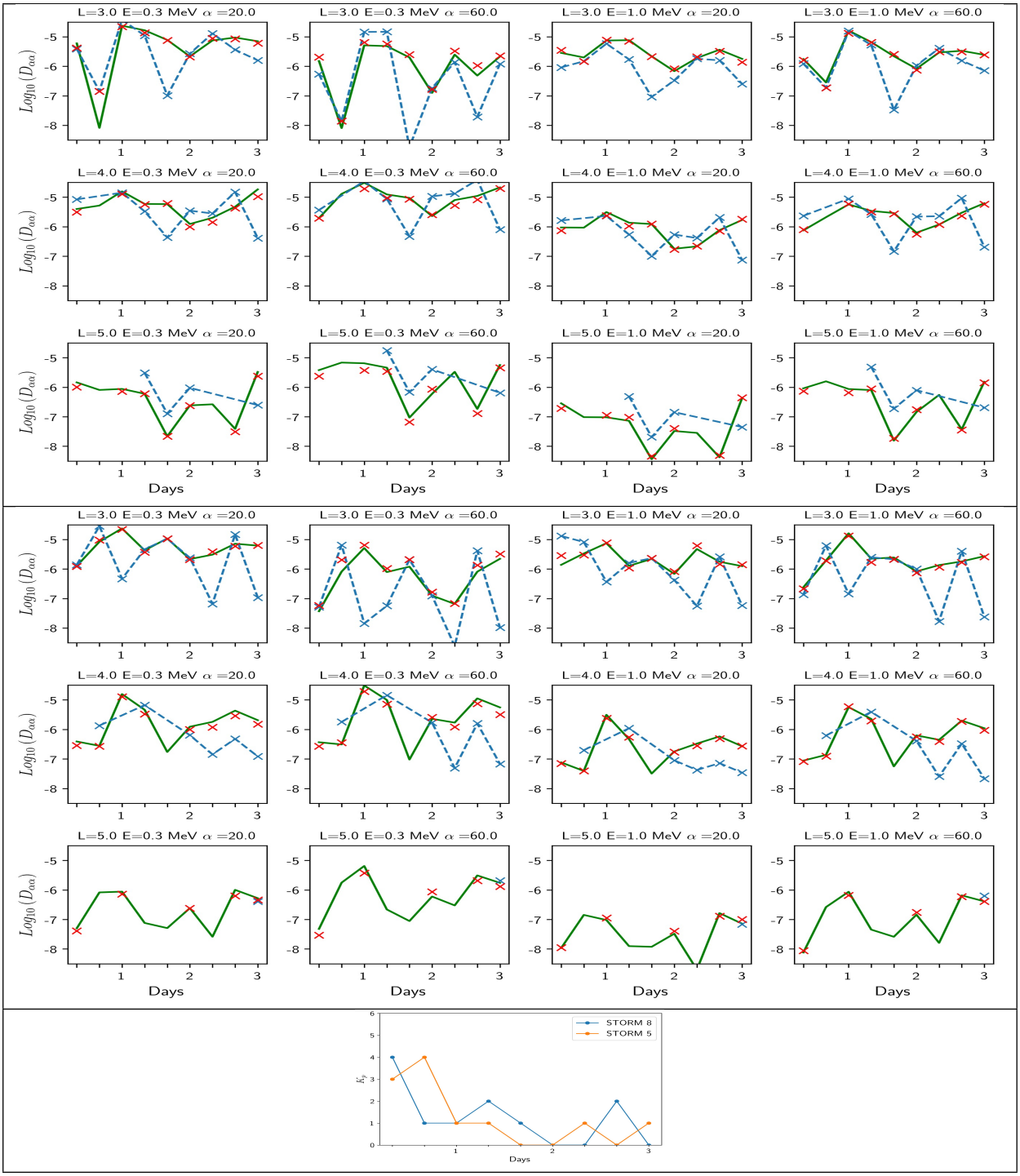

Figure 5. Pitch angle diffusion coefficients for (1-3 rows) storms 8 and (4-6 rows) storm 5. The first 6 panels show historic pitch-angle diffusion coefficient at different (L, E, α) values, with (blue crosses and lines) the raw data of event-driven coefficients, (red crosses) the averaged data (on the 32 storms at given (K_p, t, L, E, α)), and (green lines) the DNN model. The average data (in red) and the DNN model (in green) (trained on a subset of the average model) are ran from the Kp(t) sequence of each storm plotted at the bottom panel for each of the 9 temporal bins. The good agreement between red crosses and the green line shows the success of the DNN model at matching its target. Both captures levels and variations, but are not very accurate compared with the event-driven diffusion reference values in blue, showing the limits of a mean model.

At fixed (t, K_p), we see pitch angle diffusion occurs at lower energy as L increases in Figure 6 (four top rows). At low L-shell $(L < 3)$, we see a wide region of negligible diffusion in the (E, α) plane. This region of no interaction is due to the first cyclotron harmonic that does not reach pitch angles higher than the loss cone pitch angle (Ripoll et al., 2017). The DNN model has thus to learn more very low values at low L-shell. This absence of pitch angle diffusion explains why electrons are not scattered out by hiss waves and remain trapped at low L-shell in the inner belt. With the storms compressing the plasmapause, the model allows to see better if there is more effect at low L-shell. Figure 6 shows diffusion is non negligible above $\simeq$ 700 keV at L=2 and becomes stronger for active conditions (K_p=5 at t=1, first row and third column) when hiss power is localized deep inside the plasmasphere. For Landau diffusion (pitch angle above 80°) of electron below 300 keV, we notice a transition between significant Landau diffusion and an absence of diffusion for the highest pitch angle (above 85°) at K_p=3 and t=1 day, which is likely the DNN model reaching its limit. We will come back on this negative feature in the next section.

At higher L shell ($L \geq 3$) and fixed energy, the minimum pitch angle diffusion occurs between first cyclotron harmonic and the Landau (n = 0) harmonic (e.g., between $\alpha = 75$ at L = 4, E = 200 keV, K_p =3, and t=1). At fixed L shell, the maximal pitch angle diffusion from cyclotron harmonics occurs at higher energy as pitch angle increases. The sharp gradients that occur for given (L, E, α) values in the region of transition between Landau and cyclotron resonance reduces at L increases, but it remains a region of possible errors as commented in the previous section for L=3, E=0.3 MeV, and α =60 in the third row of Figure 5.

One could wonder why the diffusion at L = 4 and $K_p = 5$ is negligible at t=1. This is due to the fact that for such active condition the center of the plasmasphere where hiss are dominant (e.g.Malaspina et al. 2018) is located at lower L-shell, while L=4 is in a region of minimal hiss activity, likely in the vicinity of the plasmapause (if beyond, the wave would not be defined and the diffusion would be null). Further investigation in section 4.1 and Figure 7 will show that there exists only once case of storm having K_p=5 and t=1 so that the mean DNN model has learnt the solution shown in Figure 6 (two top rows and third column) from a single storm event. As interesting is the absence of storms with K_p=5 at t=2 days (cf. Figure 7) so that the model is extrapolating with respect to K_p in Figure 6(two top rows and fourth column). At t=2 days, the model statistically predicts some waves with some power due to the fact that likely the plasmasphere has often recovered to above L=4 at that time, bringing some hiss power. We understand the model could learn such behavior from the data. But would that be occurring in reality if K_p was still as high as K_p=5 on the second day of a HSS storm? We cannot tell from the current data.

Looking at fixed (L, E, α) values in Figure 6 (two bottom rows), we see any storm can be represented by its evolving path in the (t, K_p) space, with possibly great differences from one time to another although each storm belongs to the same kind. There is a large variability of pitch angle diffusion coefficients with respect to time looking at a horizontal line of fixed K_p. The diversity of the wave and plasma conditions leads to decay rates varying by orders of magnitude and although the K_p indices are the same. This contributes to explain why storms can be so different from one event to the other (e.g. Reeves et al. 1998). This brings the question of the time resolution of K_p (here 8 hours) and the pertinence of this index when considered as the only parameter. The MLT location of all the observations could also explain the differences. Time plays a crucial role in the solution (cf. the discussion on the interpretation of time in section 4.1), while diffusion coefficients do not depend on time in most common space weather simulations (e.g. Cervantes et al. 2020) in which only K_p remains in both the wave models and the diffusion coefficients (sometime even in the absence of the L-shell dependence (e.g. Zhu et al. 2019). The variability of the wave parameters calls for

the use of at least two geomagnetic indices or one geomagnetic index and another relevant parameter (here, directly time).

For a given (L,E), we see in Figure 6 (two bottom rows) the pattern and shape in at fixed (E, α) is roughly conserved while the levels changes. This is true because the solution is presented at not too low L-shell ($L \leq 3$) such that the region of minimal diffusion at moderately high pitch angle between the Landau and cyclotron resonance is narrower than at lower L-shell (Ripoll et al. (2017)). Nevertheless, there exist regions in the (E, α) with shapes and variations that differ from the main general trend, as, for instance, illustrated in Figure 6 (two top rows).

Further exploration of pitch angle diffusion during HSS events is discussed in Ripoll et al. (2022) and, in particular, the variability of diffusion within a same Kp index bin. This exploration of the DNN model leads us to look at which diffusion is predicted by the model during sustained HSS yet unobserved.

4 DISCUSSION

4.1 Average vs. Event-driven models

The number of storms for each activity (K_p, t) is represented in Figure 7. The specificity of storms (e.g. Reeves et al. 1998) appears clearly with a few or none events for some combinations of (K_p, t). For instance, there is no HSS storm that have a mean $K_p = 0$ within the 8 first hours. However, there is one HSS storm (over 32) for which K_p=1 occurs within the second period of 8 hours of the storm. In great majority, HSS storms have a mean K_p index of K_p=4 during the first 8 hours. 2.6 days after the storm 70% of HSS storms (22 over 32) have K_p between 1 and 2, indicative of a quite fast recovery. We also see that averages are made at fixed K_p on a maximum of 16 storms (over 32) at best for a single (K_p, t). This maximum is reached at K_p=4 in the first temporal 8 hour bin (t=0.33). The second bin with the largest number of data is ($t = 2.3$ day, K_p=2) with 14 storms. The largest spread in K_p is for the 2nd day with 5 to 9 storms in each of the K_p=0,1,2,3 bin. We have only 3 HSS storms reaching $K_p = 6$, each at 3 different times. One of them has $K_p = 6$ within the first 8 hours. Figure 7 also shows the most probable activity history of HSS, which is K_p=4, 3, 3, 2, 1, 3, 2, 2, 1. This is quite the activity of storm 12 for which we confirm we have good agreement between the event-driven diffusion coefficients and the average models (DNN and data) (not shown but similar to the results of storm 8 in Figure 5). The most probable activity history of HSS shows interestingly a main decay followed by a second milder peak of activity (with a mean Kp reaching Kp=3 again) after 48 hours. This second peak is then followed by a decay to quite times within the next 24 hours.

As we see that the error is caused by the use of averages, the immediate question arising is why averaging when making the DNN model? This is necessary here because of the way our problem is defined. If one wanted to learn directly from the individual diffusion coefficients of the 32 storms, the problem becomes multi-valued and cannot be treated by any machine learning method (unless one DNN model is done for each storm at each time, which asks then the question on how to aggregate n DNN models together). For a given (t, L, E, α), or a given (K_p, L, E, α) we found there exist multiple values of the diffusion coefficient $D_{\alpha\alpha}$. We can solve this issue by two ways: either by using more input parameters, or by averaging data. The K_p-only model is too rough and causes too much error as we will discuss next and thus $D_{\alpha\alpha}(t, K_p, L, E, \alpha)$ was retained. Here, time could be interpreted as representing any other geomagnetic index (or some global measure of them). Similarly, one could have use 2 (e.g. Dst and K_p) or 3 (or more) geomagnetic indices and their history ($Dst^* = max_{24hours}(Dst(t))$, $Kp^* = max_{24hours}(K_p(t))$) or characteristic quantities (such as solar wind velocity, dynamic pressure, etc) so that the problem becomes single valued, without

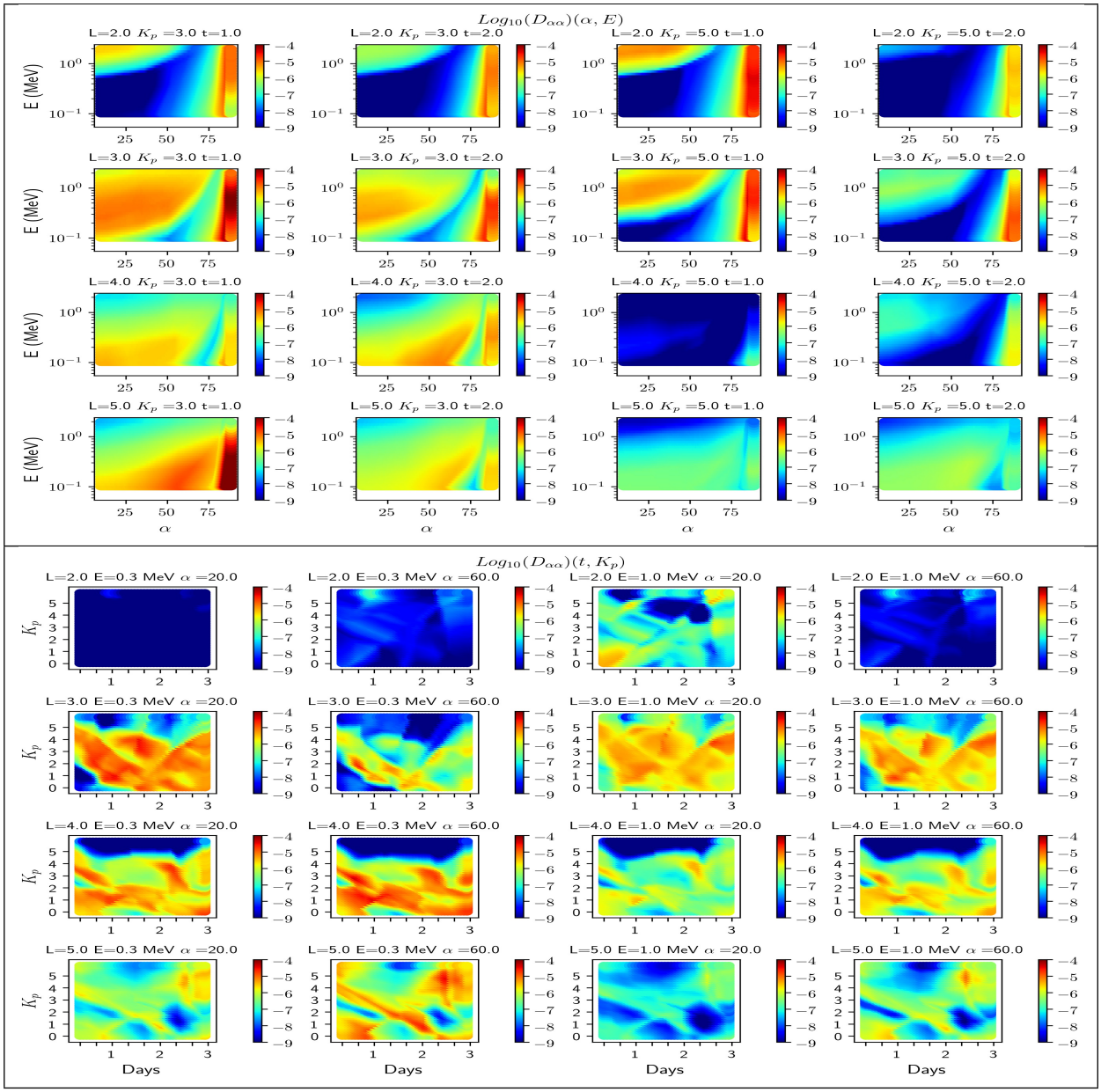

Figure 6. The DNN model of $(D_{\alpha\alpha})$ (Log_{10} of s^{-1}) in the (top) $(\alpha,\ E)$ plane at fixed (L, K_p, t) and (bottom) (t, K_p) plane at fixed $(L,\ E,\ \alpha)$.

averaging. In principle, one could also use all wave parameters as entry parameters of the unitary diffusion coefficients $D_{\alpha\alpha}(t, L, E, \alpha)$ since they were used for the generation of the single diffusion coefficients. In that case, the complexity of merging and coupling correctly various complex database together becomes an issue. Another is the knowledge of predicted wave parameters in order to use them in the model (as they are yet non unknown). Adding parameters, we reduce the possibility of encountering prohibitive multi-valued solutions and we expect it will improve the accuracy of single events.

There are still in turn 3 drawbacks to increase the data size that can alter accuracy, in particular if too many parameters were chosen. First, it increases the problem dimensions, thus the numerical cost, which should not be a problem for methods such as neural networks. Machine learning methods relying on solving for a linear system (such as the RBF method) become however unusable with too large matrices.

Dimensionality is an issue for methods that require the computation of geometrical distances, as KNN, and methods that solve for a linear system as RBF. The DNN method does not suffer from this issue and has been used in problems with hundreds and thousands of different input features. Second, there will be a larger domain in the parameter space with sparse data that will cause loss of accuracy in the region of rare occurrence. Third, increasing too much the dimension can cause over fitting of the problem, in the sense that the model loses its ability to be general and represents new events.

When going to more input variables, there is also a trade-off to find between the expected model accuracy and the variability we do not want to keep in the model, such as the dispersion caused by some measurements or very specific geophysical parameters that may be spurious. This trade-off can be quantified by the same method we use to avoid over-fitting during the training phase of the machine-learning models. The way is to start by testing models on storms that have not been seen during the training phase. When the chosen model has reached enough learning capacity, its error on these new storms will not improve, and will even grow, signifying that the learning limit has been reached.

That is why the approach we present in this article is not unique. Although we retained an approach parametrized with two paramaters, i.e. (K_p, t), the approach should be repeated for different various set of other relevant parameters, comparisons among them performed, and ultimately a choice can be made of the best parametrization reproducing the variability of the diffusion coefficients (more generally of the targeted quantity). That is why the simplest, most efficient, and accurate machine learning method has to be chosen in the first place since the method needs to be implemented quickly and replicated multiple times for different choices until eventually reaching a more definitive and more robust model.

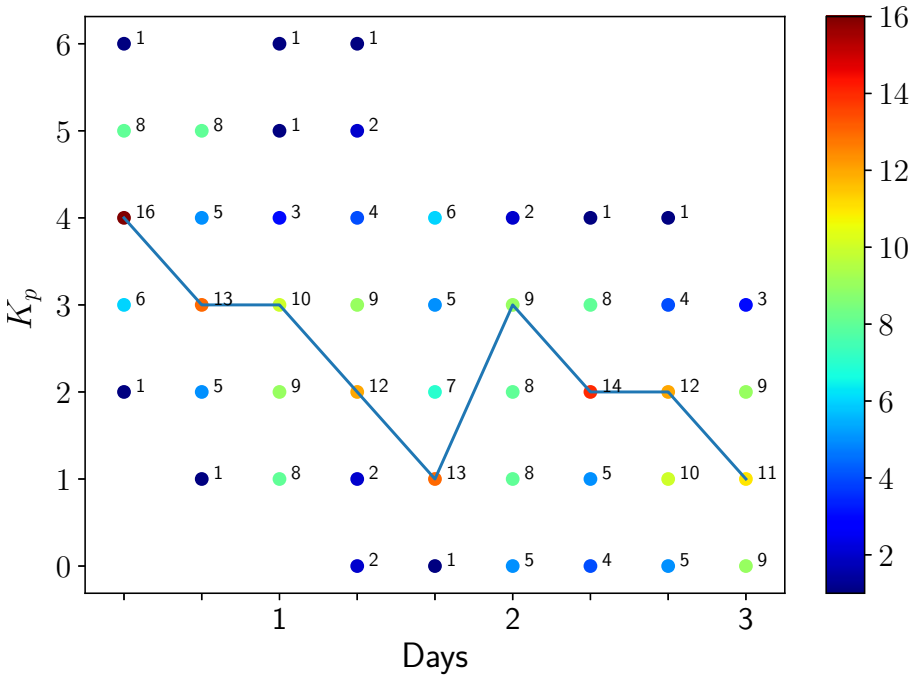

Figure 7. Numbers of storms that have K_p and t values. The number of storms in each (t,Kp) bin is written and colored and represents the number of elements averaged together per (t,Kp).

4.2 On a K_p-only model

Before the retained average model presented above, we tried a simpler model, based only on K_p, i.e. $D_{\alpha\alpha}(K_p, L, E, \alpha)$, as the modeling of pitch-angle diffusion is not time-dependent in most common space weather simulations and follows only the dynamics of a single index, such as the K_p or AE index. Interestingly, $D_{\alpha\alpha}(K_p, L, E, \alpha)$ can be obtained in three different ways: averaging the whole data DS2 over times and storms, averaging DS3_AVG over time (cf. section 2.1.4) or by averaging the machine learning model over time. The two first methods require to run through the dataset many times and to select the right data in order to perform the proper averages. These operations are prone to errors. On the other hand, averaging the DNN model is extremely seductive because immediate and simple to perform. It may contain errors due to the DNN intrinsic errors but this is compensated by the simplicity. This gives another example of positive outcome of machine learning methods.

Figure 8 shows the performance of the $D_{\alpha\alpha}(K_p,\ L,\ E,\ \alpha)$ approach for storm 8, with the DNN mean-K_p model plotted with green circle and the mean-K_p averaged data plotted with red circles (all plotted on top of the data represented in Figure 5 for illustrating the departure from the time-varying solution). First the DNN mean-K_p model and the mean-K_p averaged data agree well together which shows the success of the data assimilation by the DNN method. This also confirms a simple way to perform further global averages is to directly average the DNN model rather than to further average the data (lowering the risk of errors and simplifying greatly the task). However, both mean-K_p models gives a very rough approximation of the diffusion for a given event. They predict almost a flat curve giving only at best the central tendency. The globally low accuracy is more visible for storm 5 (which diffusion is further away to the mean diffusion) than for storm 8 (closer to the mean). This confirms the deterioration of the accuracy by any form of average; the bigger the ensemble, the higher the error.

4.3 Model limitations and future improvements

The data we use were not created specifically for this study and, as such, the discretization is not best optimized for further encapsulation by a machine learning method. The original set is too large for the herein regression in dimension 3 or 5 and the first task is a necessary strategy to reduce the amount of data. Moreover, when generating data for the purpose of machine learning modeling, an adaptive sampling strategy should be preferred. Such a method consists in optimizing at which variable in (L, E, α) the diffusion coefficient should be computed. This task is left for a future improvement of the model.

The present DNN model of HSS storms has been computed for 5 L-shells with a $\Delta L = 1$. One of the next tasks is to generalized the method to 50 L-shells covering the whole domain with $\Delta L = 0.1$. One way is to repeat the study but spread the teaching onto randomly chosen L-shells in order to keep the same resolution or to increase the sampling size, which remains possible with DNN.

Landau diffusion is the highest diffusion we see for pitch angle above $\alpha_L >\sim 80°$ in Figure 6 (top, left). At lower pitch angle, Landau diffusion is well defined but negligible (cf Mourenas and Ripoll (2012) for an approximation of α_L for a given L-shell and energy). For very large pitch angle, Landau diffusion is strong almost everywhere in the (L, E) plane, but this strong diffusion is surrounded by very weak diffusion outside $[\alpha_L, 90 - \epsilon]$, which traps and diffuses the particle within that pitch angle range. Only coupled energy-pitch angle diffusion effects can then change the electron pitch angle outside of that range (Albert et al., 2016). The region of Landau diffusion is a region with a distinct behavior that requires particular attention and can cause the DNN network to make higher local errors (as discussed previously). There can be various strategy to avoid that difficulty. One can either choose to generate two distinct DNN model, one

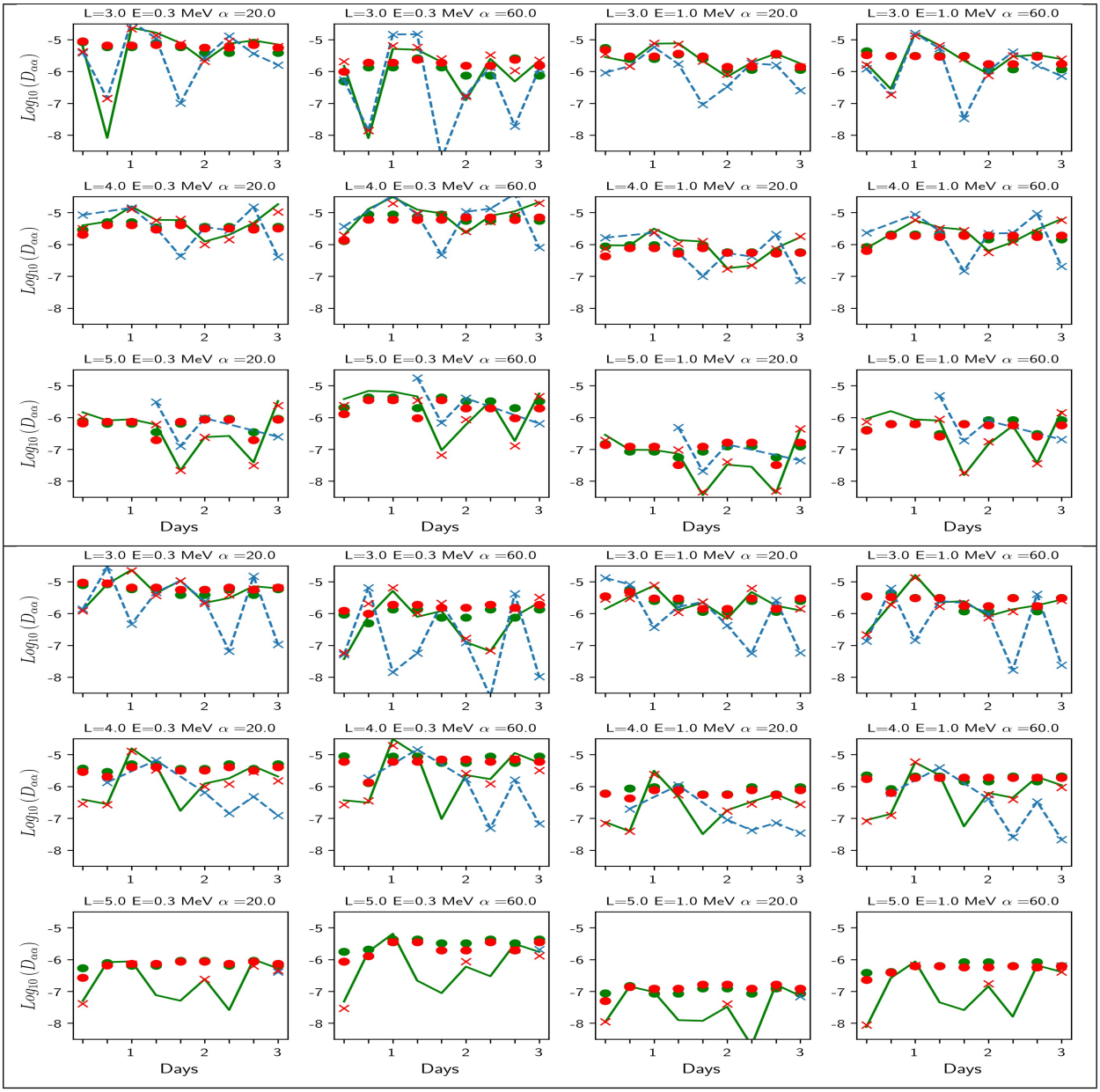

Figure 8. History of the storms 8 and 5. We plot here the same results as in Figure 5 (blue: raw data, red crosses: (K_p, t, L, E, α) average model, green crosses: (K_p, t, L, E, α) DNN model) to which we added circles obtained from averaging in time either the average data (red) or directly the DNN model (green); it produces a K_p-dependent (only) model. The DNN model approaches well its target (the average data) but both have a degraded accuracy compared to the event-driven model (in blue), particularly visible for storm 5 which diffusion coefficients depart significantly from the average.

for low and moderate pitch angles (which has the effect to focus on cyclotron resonance) and the other for larger ones (above $\alpha_L >\sim 80°$ where Landau generally occurs). This strategy can be tricky because the exact position of the Landau resonance varies also with the wave and density properties (Mourenas and Ripoll, 2012) leading to a dependence with (t, K_p, L, E). The better and simpler strategy, which our study brings, is to separate the sum of the n-cyclotron harmonics of the diffusion coefficient from the Landau harmonic ($n = 0$) when the diffusion coefficients are computed and to store both. Then, it is straightforward and more accurate to build a DNN model for each of the two components: one for the

n-cyclotron and one for the Landau component. The full model is then made by the sum of both models. The only drawback is the increase of the memory storage by a factor 2.

Finally, machine learning models provide a wide and continuous model in a high dimensional space, which can produce extrapolation and surprising results (right or wrong) in particular for rare events and in the various high-dimensional corners of the model. These solutions always require for verification to go back to the database and to explore it more and more to the point of knowing (or trying to know) the data in all its aspects. This is often very time consuming, if not practically impossible, even if facilitated by the machine learning method in use. These difficulties call for reliable and robust testing methods and metrics to be able to rely more and more on the machine learning method with less and less verification of the database. In this work, even though the DNN model has shown a good accuracy, we do not think we have yet reached this level as, for instance, there are some remaining issues due to strong gradients (e.g. associated to Landau diffusion) or there is no possibility to verify and validate the behavior of the model for special configuration (e.g. low K_p in the first time of the HSS storm). The second point may call for using a given mean model simultaneously with its variance, which signifies using DNN that propagate the distribution of the data. A mean answer would be given with a confidence index based on the variance. The generation of DNN-based median, quartile, and standard deviation of the diffusion coefficients is thus a promising next step to help selecting a given model. A second important application brought by the knowledge of both the mean and variance is the ability to perform with them uncertainty analysis of Fokker Planck simulation (e.g. Camporeale et al. 2019) and better establish the variability caused by storms and better rank the best possible scenarios for given conditions.

5 CONCLUSIONS

In this work, we consider 8 nonparametric methods of machine learning based on local evaluation (k-nearest neighbors and kernel regression), tree-based methods (regression tree, bagging and random forest), neural networks and function approximations (Radial basis and splines). With them, we derive machine learning based models of event-driven diffusion coefficients first for the storm of March 2013 associated to high-speed streams. We present an approach that exhibits some selected properties of the machine learning models in order to select the best method for our problem among the 8 methods. The approach is based on 3 diagnostics: compute the main global metrics (including mean, median, minimum, maximum, standard deviation and quartiles errors) at various resolution of the database, generate violin plots for analyzing the error distribution, and compute the correlation of each method with the other to enlighten their differences and exhibit the main families. We find that neural networks (DNN), radial basis functions and splines methods performed the best for this storm, with DNN retained for the next steps of the study.

We then use the diffusion coefficients computed from 32 high-speed storms in order to build a statistical event-driven diffusion coefficients that is embedded within the retained DNN model. This is the first model of that kind for two reasons. First the machine learning model encapsulates the statistical event-driven diffusion coefficients. Second, this is the first statistical diffusion coefficients made from averaging event-driven coefficients. The common approach is to rather build statistical wave and plasma properties and to compute single diffusion coefficients from them.

The statistics of the event-driven diffusion coefficients is based on the mean with a double parametrization in epoch time and K_p. The double parametrization is chosen to keep both the strength of the storm and follow its history through epoch time. In comparison, a K_p-only model is found too inaccurate compared with specific event-driven diffusion coefficients (by 1 to 2 orders of magnitudes depending on the event).

The machine learning model step is made for greatly facilitating the use of the mean model, for instance, in providing a continuous solution in a high dimensional space (e.g. (t, L, E, α, K_p). We find the DNN model does not entail any issue to interpolate the averaged model and reproduces quite perfectly its target. Some small deviations are found at very high pitch angle for Landau resonance for which we propose a future solution to by-pass this difficulty. We then use the DNN mean model to analyze how mean diffusion coefficients behave compared with individual ones. We find a poor performance of any mean models compared with individual events, with mean models computing the general trend at best. Degradation of the accuracy of mean diffusion coefficients comes for the large variance of event-driven diffusion coefficients. Mean models can easily deviate by 2 orders of magnitude. This is shown to occur, for instance, in region of strong gradients of the diffusion coefficients, basically delimited by the edge of the first cyclotron resonance in the (E, α) plane.

The strength of the DNN approach is the simplicity of performing comparisons since the model delivers continuous map of the solution with a simple numerical subroutine for a problem with 5 to 6 dimensions here. This is illustrated by model exploration provided in section 3.2.3. Plotting diffusion coefficients in the (t, K_p) plane, for instance, shows a wide variety of solutions, contributing to explain why storms can be so different from one event to the other.

Machine learning methods and the easily accessible numerical procedures that favor their use have a wide potential for the type of problems we presented, whether it is for manipulating, interpolating, representing, or for analyzing huge database of event-driven diffusion coefficients and, more generally, database of diffusion coefficients combined with the main parameters used to compute them, such as plasma density and wave parameters. A inherent drawback is the human involvement required to analyze these huge database in order to potentially identify regions of model deviance or model breakthrough.

The DNN method that is proposed here has the advantage to be extended to more parameters characterizing storms (including OMNI solar wind and geomagnetic index data), which should improve the accuracy and predictability of global models. DNN can similarly be used to derive DNN-based median, quartile and standard deviation of the diffusion coefficients. With them, one can perform uncertainty analysis of Fokker Planck simulation and better establish the variability caused by storms and better rank the best possible scenarios for given conditions. We expect this approach to take more importance in the coming years.

CONFLICT OF INTEREST STATEMENT

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

AUTHOR CONTRIBUTIONS

All authors contributed to the manuscript and the data analysis, with a synthesis made by GK. Data were constructed by JFR and processed by GK. JFR and EC brought their expertise in geophysics. SH, AF, MM, EC and GK in machine learning. Models were made by SH, AF, MM, except for the radial basis model (EC) and the deep neural network learning model (GK). GK and JFR together wrote the first draft of the manuscript. JFR performed the physical analysis. All authors contributed to manuscript revision, read, and approved the submitted version.

FUNDING

EC is partially funded by the National Aeronautics and Space Administration under grant 80NSSC20K1580 "SWQU: Ensamble Learning for Accurate and Reliable Uncertainty Quantification"

DATA AVAILABILITY STATEMENT

Publicly available datasets were used in this study. EMFISIS data was obtained from `https://emfisis.physics.uiowa.edu/Flight/`. Data are also available at NASA CDAWeb `https://cdaweb.gsfc.nasa.gov/index.html/`.

Datafiles and both RBF and KerReg models are hosted on `https://github.com/ML-Space-Weather/PADiffusion-HSS`. For DNN models, we used DJINN package `https://github.com/LLNL/DJINN`. It is based on `scikit-learn` and `tensorflow` library. The tree-based and local evaluation methods are based on `sklearn` library. We use both the `tree` and the `neighbors` functions for Regression tree and KNN, respectively. We use both `BaggingRegressor` and `RandomForestRegressor` functions of `ensemble` module (sklearn.ensemble) of `sklearn`, respectively. And we use `KFold` function of `model_selection` module from `sklearn` to perform K-fold cross validation in selecting the hyperparameters of the methods. Lastly, the Thin Plate Spline regression was implemented in `R` software using `Tps` function of `fields` library `https://cran.r-project.org/package=fields`.

REFERENCES

Albert, J. M. (1994). Quasi-linear pitch angle diffusion coefficients: Retaining high harmonics. Journal of Geophysical Research: Space Physics 99, 23741–23745. doi:https://doi.org/10.1029/94JA02345

Albert, J. M. (1999). Analysis of quasi-linear diffusion coefficients. Journal of Geophysical Research: Space Physics 104, 2429–2441. doi:https://doi.org/10.1029/1998JA900113

Albert, J. M. (2005). Evaluation of quasi-linear diffusion coefficients for whistler mode waves in a plasma with arbitrary density ratio. Journal of Geophysical Research: Space Physics 110. doi:https://doi.org/10.1029/2004JA010844

Albert, J. M. (2008). Efficient approximations of quasi-linear diffusion coefficients in the radiation belts. Journal of Geophysical Research: Space Physics 113. doi:https://doi.org/10.1029/2007JA012936

Albert, J. M., Starks, M. J., Horne, R. B., Meredith, N. P., and Glauert, S. A. (2016). Quasi-linear simulations of inner radiation belt electron pitch angle and energy distributions. Geophysical Research Letters 43, 2381–2388. doi:https://doi.org/10.1002/2016GL067938

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician 46, 175–185. doi:10.1080/00031305.1992.10475879

Baker, D. N., Jaynes, A. N., Li, X., Henderson, M. G., Kanekal, S. G., Reeves, G. D., et al. (2014). Gradual diffusion and punctuated phase space density enhancements of highly relativistic electrons: Van allen probes observations. Geophysical Research Letters 41, 1351–1358. doi:https://doi.org/10.1002/2013GL058942

Bala, R. and Reiff, P. (2012). Improvements in short-term forecasting of geomagnetic activity. Space Weather 10

Breiman, L. (1996). Bagging predictors. Machine Learning 24, 123–140. doi:10.1007/BF00058655

Breiman, L. (2001). Random forests. Machine Learning 45, 5–32. doi:10.1023/A:1010933404324

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). Classification and Regression Trees (CA: Wadsworth and Brooks/Cole Advanced Books and Software)

Breneman, A. W., Halford, A., Millan, R., McCarthy, M., Fennell, J., Sample, J., et al. (2015). Global-scale coherence modulation of radiation-belt electron loss from plasmaspheric hiss. Nature 523, 193–195. doi:10.1038/nature14515

Camporeale, E. (2019). The challenge of machine learning in space weather: Nowcasting and forecasting. Space Weather 17, 1166–1207. doi:https://doi.org/10.1029/2018SW002061

Camporeale, E., Chu, X., Agapitov, O., and Bortnik, J. (2019). On the generation of probabilistic forecasts from deterministic models. Space Weather 17, 455–475

Cervantes, S., Shprits, Y. Y., Aseev, N. A., and Allison, H. J. (2020). Quantifying the effects of emic wave scattering and magnetopause shadowing in the outer electron radiation belt by means of data assimilation. Journal of Geophysical Research: Space Physics 125, e2020JA028208. doi:https://doi.org/10.1029/2020JA028208

Cesaroni, C., Spogli, L., Aragon-Angel, A., Fiocca, M., Dear, V., De Franceschi, G., et al. (2020). Neural network based model for global total electron content forecasting. Journal of Space Weather and Space Climate

Chu, X., Bortnik, J., Li, W., Ma, Q., Denton, R., Yue, C., et al. (2017a). A neural network model of three-dimensional dynamic electron density in the inner magnetosphere. Journal of Geophysical Research: Space Physics 122, 9183–9197. doi:https://doi.org/10.1002/2017JA024464

Chu, X. N., Bortnik, J., Li, W., Ma, Q., Angelopoulos, V., and Thorne, R. M. (2017b). Erosion and refilling of the plasmasphere during a geomagnetic storm modeled by a neural network. Journal of Geophysical Research: Space Physics 122, 7118–7129. doi:https://doi.org/10.1002/2017JA023948

Cybenko, G. (1989). Approximations by superpositions of sigmoidal functions. Mathematics of Control, Signals ans Systems 2

Duchon, J. (1977). Splines minimizing rotation-invariant semi-norms in sobolev spaces. In Construction Theory of Functions of Several Variables (Berlin: Springer)

Fix, E. and Hodges, J. (1951). Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties (Report). Tech. rep.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning (The MIT Press)

Green, P. J. and Silverman, B. W. (1994). Nonparametric regression and generalized linear models: a roughness penalty approach (United Kingdom: Chapman and Hall)

Gruet, M. A., Chandorkar, M., Sicard, A., and Camporeale, E. (2018). Multiple-hour-ahead forecast of the dst index using a combination of long short-term memory neural network and gaussian process. Space Weather 16, 1882–1896

Géron, A. (2017). Machine Learning avec Scikit-Learn (Dunod)

Hastie, T., Tibshirani, R., and Friedman, J. (2009). The Elements of Statistical Learning (Springer)

Ho, T. K. (1995). Random decision forests. In Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1 (USA: IEEE Computer Society), 278

Horne, R. B., Kersten, T., Glauert, S. A., Meredith, N. P., Boscher, D., Sicard-Piet, A., et al. (2013). A new diffusion matrix for whistler mode chorus waves. Journal of Geophysical Research: Space Physics 118, 6302–6318. doi:https://doi.org/10.1002/jgra.50594

Humbird, K., Peterson, J., and McClarren, R. (2019). Deep neural network initialization with decision trees. IEEE Transactions on neural networks and learning systems 30

Kletzing, C. A., Kurth, W. S., Acuna, M., MacDowall, R. J., Torbert, R. B., Averkamp, T., et al. (2013). The electric and magnetic field instrument suite and integrated science (emfisis) on rbsp. Space Science Reviews 179, 127–181. doi:10.1007/s11214-013-9993-6

Kluth, G., Humbird, K., Spears, B., Peterson, J., Scott, H., Patel, M., et al. (2020). Deep learning for nlte spectral opacities. Physics of plasma 27

Kurth, W. S., De Pascuale, S., Faden, J. B., Kletzing, C. A., Hospodarsky, G. B., Thaller, S., et al. (2015). Electron densities inferred from plasma wave spectra obtained by the waves instrument on van allen probes. Journal of Geophysical Research: Space Physics 120, 904–914. doi:https://doi.org/10.1002/2014JA020857

Li, W., Ma, Q., Thorne, R. M., Bortnik, J., Kletzing, C. A., Kurth, W. S., et al. (2015). Statistical properties of plasmaspheric hiss derived from van allen probes data and their effects on radiation belt electron dynamics. Journal of Geophysical Research: Space Physics 120, 3393–3405. doi:https://doi.org/10.1002/2015JA021048

Li, Z., Hudson, M., Jaynes, A., Boyd, A., Malaspina, D., Thaller, S., et al. (2014). Modeling gradual diffusion changes in radiation belt electron phase space density for the march 2013 van allen probes case study. Journal of Geophysical Research: Space Physics 119, 8396–8403. doi:https://doi.org/10.1002/2014JA020359

Liemohn, M. W., McCollough, J. P., Jordanova, V. K., Ngwira, C. M., Morley, S. K., Cid, C., et al. (2018). Model evaluation guidelines for geomagnetic index predictions. Space Weather 16, 2079–2102. doi:https://doi.org/10.1029/2018SW002067

Linty, N., Farasin, A., Favenza, A., and Dovis, F. (2018). Detection of gnss ionospheric scintillations based on machine learning decision tree. IEEE Transactions on Aerospace and Electronic Systems 55, 303–317

Loridan, V., Ripoll, J.-F., Tu, W., and Scott Cunningham, G. (2019). On the use of different magnetic field models for simulating the dynamics of the outer radiation belt electrons during the october 1990 storm. Journal of Geophysical Research: Space Physics 124, 6453–6486. doi:https://doi.org/10.1029/2018JA026392

Lyons, L. R. and Thorne, R. M. (1973). Equilibrium structure of radiation belt electrons. Journal of Geophysical Research (1896-1977) 78, 2142–2149. doi:https://doi.org/10.1029/JA078i013p02142

Lyons, L. R., Thorne, R. M., and Kennel, C. F. (1972). Pitch-angle diffusion of radiation belt electrons within the plasmasphere. Journal of Geophysical Research (1896-1977) 77, 3455–3474. doi:https://doi.org/10.1029/JA077i019p03455

Ma, Q., Li, W., Bortnik, J., Thorne, R. M., Chu, X., Ozeke, L. G., et al. (2018). Quantitative evaluation of radial diffusion and local acceleration processes during gem challenge events. Journal of Geophysical Research: Space Physics 123, 1938–1952. doi:https://doi.org/10.1002/2017JA025114

Malaspina, D. M., Ripoll, J.-F., Chu, X., Hospodarsky, G., and Wygant, J. (2018). Variation in plasmaspheric hiss wave power with plasma density. Geophysical Research Letters 45, 9417–9426. doi:https://doi.org/10.1029/2018GL078564

Mauk, B. H., Fox, N. J., Kanekal, S. G., Kessel, R. L., Sibeck, D. G., and Ukhorskiy, A. (2013). Science objectives and rationale for the radiation belt storm probes mission. Space Science Reviews 179, 3–27. doi:10.1007/s11214-012-9908-y

McGranaghan, R. M., Mannucci, A. J., Wilson, B., Mattmann, C. A., and Chadwick, R. (2018). New capabilities for prediction of high-latitude ionospheric scintillation: A novel approach with machine learning. Space Weather 16, 1817–1846

McGranaghan, R. M., Ziegler, J., Bloch, T., Hatch, S., Camporeale, E., Lynch, K., et al. (2021). Toward a next generation particle precipitation model: Mesoscale prediction through machine learning (a case study and framework for progress). Space Weather , e2020SW002684

Meredith, N. P., Horne, R. B., Glauert, S. A., and Anderson, R. R. (2007). Slot region electron loss timescales due to plasmaspheric hiss and lightning-generated whistlers. Journal of Geophysical Research: Space Physics 112. doi:https://doi.org/10.1029/2007JA012413

Meredith, N. P., Horne, R. B., Glauert, S. A., Baker, D. N., Kanekal, S. G., and Albert, J. M. (2009). Relativistic electron loss timescales in the slot region. Journal of Geophysical Research: Space Physics 114. doi:https://doi.org/10.1029/2008JA013889

Meredith, N. P., Horne, R. B., Kersten, T., Li, W., Bortnik, J., Sicard, A., et al. (2018a). Global model of plasmaspheric hiss from multiple satellite observations. Journal of Geophysical Research: Space Physics 123, 4526–4541. doi:https://doi.org/10.1029/2018JA025226

Meredith, N. P., Horne, R. B., Kersten, T., Li, W., Bortnik, J., Sicard, A., et al. (2018b). Global model of plasmaspheric hiss from multiple satellite observations. Journal of Geophysical Research: Space Physics 123, 4526–4541. doi:https://doi.org/10.1029/2018JA025226

Millan, R. M., McCarthy, M. P., Sample, J. G., Smith, D. M., Thompson, L. D., McGaw, D. G., et al. (2013). The balloon array for rbsp relativistic electron losses (barrel). Space Science Reviews 179, 503–530. doi:10.1007/s11214-013-9971-z

Millan, R. M., Ripoll, J.-F., Santolík, O., and Kurth, W. S. (2021). Early-time non-equilibrium pitch angle diffusion of electrons by whistler-mode hiss in a plasmaspheric plume associated with barrel precipitation. Frontiers in Astronomy and Space Sciences 8. doi:10.3389/fspas.2021.776992

Mourenas, D. and Ripoll, J.-F. (2012). Analytical estimates of quasi-linear diffusion coefficients and electron lifetimes in the inner radiation belt. Journal of Geophysical Research: Space Physics 117. doi:https://doi.org/10.1029/2011JA016985

Nadaraya, E. A. (1964). On estimating regression. Theory of Probability & Its Applications 9, 141–142. doi:10.1137/1109020

Oyeyemi, E., Poole, A., and McKinnell, L. (2005). On the global model for fof2 using neural networks. Radio science 40

Pérez, D., Wohlberg, B., Lovell, T. A., Shoemaker, M., and Bevilacqua, R. (2014). Orbit-centered atmospheric density prediction using artificial neural networks. Acta Astronautica 98, 9–23

Reeves, G. D., Baker, D. N., Belian, R. D., Blake, J. B., Cayton, T. E., Fennell, J. F., et al. (1998). The global response of relativistic radiation belt electrons to the january 1997 magnetic cloud. Geophysical Research Letters 25, 3265–3268. doi:https://doi.org/10.1029/98GL02509

Reeves, G. D., Friedel, R. H. W., Larsen, B. A., Skoug, R. M., Funsten, H. O., Claudepierre, S. G., et al. (2016). Energy-dependent dynamics of kev to mev electrons in the inner zone, outer zone, and slot regions. Journal of Geophysical Research: Space Physics 121, 397–412. doi:https://doi.org/10.1002/2015JA021569

Ripoll, J.-F., Claudepierre, S. G., Ukhorskiy, A. Y., Colpitts, C., Li, X., Fennell, J. F., et al. (2020a). Particle dynamics in the earth's radiation belts: Review of current research and open questions. Journal of Geophysical Research: Space Physics 125, e2019JA026735. doi:https://doi.org/10.1029/2019JA026735. E2019JA026735 2019JA026735

Ripoll, J.-F., Denton, M., Hartley, D., Reeves, G., Malaspina, D., Cunningham, G., et al. (2020b). Scattering by whistler-mode waves during a quiet period perturbed by substorm activity. Journal of Atmospheric and Solar-Terrestrial Physics , 105471doi:https://doi.org/10.1016/j.jastp.2020.105471

Ripoll, J.-F., Kluth, G., Has, S., Fischer, A., Mougeot, M., and Camporeale, E. (2022). A neural network model of quasi-linear diffusion coefficients during high-speed streams. Proceedings of the 3rd URSI AT-AP-RASC, Gran Canaria, 29 May – 3 June 2022

Ripoll, J.-F., Loridan, V., Cunningham, G. S., Reeves, G. D., and Shprits, Y. Y. (2016a). On the time needed to reach an equilibrium structure of the radiation belts. Journal of Geophysical Research: Space Physics 121, 7684–7698. doi:https://doi.org/10.1002/2015JA022207

Ripoll, J.-F., Loridan, V., Denton, M. H., Cunningham, G., Reeves, G., Santolík, O., et al. (2019). Observations and fokker-planck simulations of the l-shell, energy, and pitch angle structure of earth's electron radiation belts during quiet times. Journal of Geophysical Research: Space Physics 124, 1125–1142. doi:https://doi.org/10.1029/2018JA026111

Ripoll, J.-F. and Mourenas, D. (2012). High-Energy Electron Diffusion by Resonant Interactions with Whistler Mode Hiss (American Geophysical Union (AGU)). 281–290. doi:https://doi.org/10.1029/2012GM001309

Ripoll, J.-F., O., S., Reeves, G., Kurth, W., Denton, M., Loridan, V., et al. (2017). Effects of whistlermode hiss waves in march 2013. Journal of Geophysical Research: Space Physics 122, e2019JA026735. doi:10.1002/2017JA024139

Ripoll, J.-F., Reeves, G. D., Cunningham, G. S., Loridan, V., Denton, M., Santolík, O., et al. (2016b). Reproducing the observed energy-dependent structure of earth's electron radiation belts during storm recovery with an event-specific diffusion model. Geophysical Research Letters 43, 5616–5625. doi:https://doi.org/10.1002/2016GL068869

Réveillé, T. (1997). Etude de mécanismes de pertes de particules dans les ceintures artificielles de van allen. Ph.D. Thesis. France: Univ. Henri Poincaré, Nancy-I.

Réveillé, T., Bertrand, P., Ghizzo, A., Simonet, F., and Baussart, N. (2001). Dynamic evolution of relativistic electrons in the radiation belts. Journal of Geophysical Research: Space Physics 106, 18883–18894. doi:https://doi.org/10.1029/2000JA900177

Santolík, O., Parrot, M., Storey, L. R. O., Pickett, J. S., and Gurnett, D. A. (2001). Propagation analysis of plasmaspheric hiss using polar pwi measurements. Geophysical Research Letters 28, 1127–1130. doi:https://doi.org/10.1029/2000GL012239

Sicard-Piet, A., Boscher, D., Horne, R. B., Meredith, N. P., and Maget, V. (2014). Effect of plasma density on diffusion rates due to wave particle interactions with chorus and plasmaspheric hiss: extreme event analysis. Annales Geophysicae 32, 1059–1071. doi:10.5194/angeo-32-1059-2014

Siciliano, F., Consolini, G., Tozzi, R., Gentili, M., Giannattasio, F., and De Michelis, P. (2021). Forecasting sym-h index: A comparison between long short-term memory and convolutional neural networks. Space Weather 19, e2020SW002589

Spasojevic, M., Shprits, Y. Y., and Orlova, K. (2015). Global empirical models of plasmaspheric hiss using van allen probes. Journal of Geophysical Research: Space Physics 120, 10,370–10,383. doi:https://doi.org/10.1002/2015JA021803

Sun, W., Xu, L., Huang, X., Zhang, W., Yuan, T., Chen, Z., et al. (2017). Forecasting of ionospheric vertical total electron content (tec) using lstm networks. In 2017 International Conference on Machine Learning and Cybernetics (ICMLC) (IEEE), vol. 2, 340–344

Takalo, J. and Timonen, J. (1997). Neural network prediction of ae data. Geophysical research letters 24, 2403–2406

Thaller, S. A., Wygant, J. R., Dai, L., Breneman, A. W., Kersten, K., Cattell, C. A., et al. (2015). Van allen probes investigation of the large-scale duskward electric field and its role in ring current formation and

plasmasphere erosion in the 1 june 2013 storm. Journal of Geophysical Research: Space Physics 120, 4531–4543. doi:https://doi.org/10.1002/2014JA020875

Thorne, R. M., Li, W., Ni, B., Ma, Q., Bortnik, J., Chen, L., et al. (2013). Rapid local acceleration of relativistic radiation-belt electrons by magnetospheric chorus. Nature 504, 411–414. doi:10.1038/nature12889

Tulunay, E., Senalp, E. T., Radicella, S. M., and Tulunay, Y. (2006). Forecasting total electron content maps by neural network technique. Radio science 41

Turner, D. L., Kilpua, E. K. J., Hietala, H., Claudepierre, S. G., O'Brien, T. P., Fennell, J. F., et al. (2019). The response of earth's electron radiation belts to geomagnetic storms: Statistics from the van allen probes era including effects from different storm drivers. Journal of Geophysical Research: Space Physics 124, 1013–1034. doi:https://doi.org/10.1029/2018JA026066

Wang, D., Shprits, Y. Y., Zhelavskaya, I. S., Effenberger, F., Castillo, A. M., Drozdov, A. Y., et al. (2020). The effect of plasma boundaries on the dynamic evolution of relativistic radiation belt electrons. Journal of Geophysical Research: Space Physics 125, e2019JA027422. doi:https://doi.org/10.1029/2019JA027422

Watson, G. S. (1964). Smooth regression analysis. Sankhyā: The Indian Journal of Statistics, Series A (1961-2002) 26, 359–372

Watt, C. E. J., Allison, H. J., Thompson, R. L., Bentley, S. N., Meredith, N. P., Glauert, S. A., et al. (2021). The implications of temporal variability in wave-particle interactions in earth's radiation belts. Geophysical Research Letters 48, e2020GL089962. doi:https://doi.org/10.1029/2020GL089962

Wygant, J. R., Bonnell, J. W., Goetz, K., Ergun, R. E., Mozer, F. S., Bale, S. D., et al. (2013). The electric field and waves instruments on the radiation belt storm probes mission. Space Science Reviews 179, 183–220. doi:10.1007/s11214-013-0013-7

Zhelavskaya, I. S., Shprits, Y. Y., and Spasojevic, M. (2018). Chapter 12 - reconstruction of plasma electron density from satellite measurements via artificial neural networks. In Machine Learning Techniques for Space Weather, eds. E. Camporeale, S. Wing, and J. R. Johnson (Elsevier). 301 – 327. doi:https://doi.org/10.1016/B978-0-12-811788-0.00012-3

Zhelavskaya, I. S., Shprits, Y. Y., and Spasojević, M. (2017). Empirical modeling of the plasmasphere dynamics using neural networks. Journal of Geophysical Research: Space Physics 122, 11,227–11,244. doi:https://doi.org/10.1002/2017JA024406

Zhelavskaya, I. S., Spasojevic, M., Shprits, Y. Y., and Kurth, W. S. (2016). Automated determination of electron density from electric field measurements on the van allen probes spacecraft. Journal of Geophysical Research: Space Physics 121, 4611–4625. doi:https://doi.org/10.1002/2015JA022132

Zhu, H., Shprits, Y. Y., Spasojevic, M., and Drozdov, A. Y. (2019). New hiss and chorus waves diffusion coefficient parameterizations from the van allen probes and their effect on long-term relativistic electron radiation-belt verb simulations. Journal of Atmospheric and Solar-Terrestrial Physics 193, 105090. doi:https://doi.org/10.1016/j.jastp.2019.105090